DOCUMENT RESUME

ED 278 378 IR 012 491

TITLE	ADCIS Conference Proceedings. Computer-Based Instruction (28th, Washington, D.C., November 10-13, 1986).
INSTITUTION	Association for the Development of Computer-based Instructional Systems.
PUB DATE	Nov 86
NOTE	515p.; For the proceedings of the 27th conference, see ED 272 177.
PUB TYPE	Collected Works - Conference Proceedings (021) -- Viewpoints (120) -- Reports - Descriptive (141)
EDRS PRICE	MF02/PC21 Plus Postage.
DESCRIPTORS	Abstracts; *Computer Assisted Instruction; Computer Assisted Testing; Computer Simulation; *Computer Software; Database Management Systems; Elementary Secondary Education; Expert Systems; Higher Education; Interactive Video; *Media Research; *Microcomputers; Program Descriptions; Programing Languages; *Teaching Methods; Technological Advancement
IDENTIFIERS	*Association for Devel Computer Based Instr Systs; Pilot Programing Language; PLATO; Spreadsheets

ABSTRACT

This volume is divided into full formal paper manuscripts and 200-word abstracts of presentations that were made without submitting formal papers. The 73 papers presented in full represent recent research and applications in the field of computer-based instruction. They are organized into 13 sections by special interest group: (1) Computer-Based Training; (2) Elementary, Secondary, Junior College Educators; (3) Health Education; (4) Home Economics Consortium; (5) Interagency Group for Computer-Based Training; (6) Implementation; (7) Mini-Micro Computer Users; (8) Management Issues; (9) Music Educators Consortium; (10) PILOT Users Group; (11) PLATO Users Group; (12) Interactive Video-Audio; and (13) Theory and Research. More than 60 200-word abstracts for additional presentations are also provided, as are lists of the Association for the Development of Computer-Based Instructional Systems (ADCIS) Board of Directors and Special Interest Group Council, and an author index. (DJR)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

BEST COPY AVAILABLE

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
☒ This document has been reproduced as received from the person or organization originating it.
☐ Minor changes have been made to improve reproduction quality.
● Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

ED278378

IR012491

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
J.J.Hirshbuhl

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

28th ADCIS CONFERENCE PROCEEDINGS

Association for the Development of Computer-Based Instructional Systems

WASHINGTON, D.C. - CRYSTAL CITY
NOVEMBER 10 thru 13, 1986

Copyright ADCIS 1986
ADCIS, MILLER HALL 409
WESTERN WASHINGTON UNIVERSITY
BELLINGHAM, WASHINGTON 98225

FOREWORD

ADCIS is proud to present these papers and abstracts from the 28th International Conference, the largest conference in the world on the use of computers in education and training. By its publication, this document makes a major contribution to the advancement of the education and training professions. This is a major function of the association.

The work presented by this Proceedings represents hundreds of thousands of staff hours and millions of dollars invested. As you use this document and meet the many authors, acknowledge their efforts and expertise, and let them know they are appreciated. It is their leadership and perserverance that enables us to advance quickly in improving the quality and efficiency of learning for everyone.

Glenn E. Head
Program Chairperson

PRESIDENT'S MESSAGE

The Proceedings of the 28th International ADCIS Conference are the largest and most diverse that have been published in the history of our Association. This publication is divided into full formal paper manuscripts and 200-word abstracts of presentations that were made without submitting formal papers. We have instituted this expanded format to better record the knowledge and information that is shared at the Annual Meetings and to provide a more comprehensive body of literature related to the theory and practical implementation of computer-based education and training.

This year's Conference Program was the largest that ADCIS has ever had the privilege of offering to its membership and to other computer-based instruction professionals. As the contents of the Proceedings will support, no matter what your field of expertise or your area of interest, you will find papers and presentations that will prove to be beneficial and to expand your general knowledge about computer-based instruction.

These Proceedings are published for the benefit of computer-based education and training professionals everywhere. Your comments and suggestions to improve this publication are much appreciated.

Ronald C. Comer
President

ADCIS gratefully acknowledges the cooperation of Western Washington University Art Department in the design of the cover for the Conference Program and Proceedings for the last two years. This year's cover was done by Al Yates under the direction of Professor Madge Gleeson.

ADCIS BOARD OF DIRECTORS

President
Ron Comer
The Ohio State University

Program Chairperson
Glenn Head
Instructional Communications

Presession Chair
Dale Brubeck
The Ohio State University

JCBI Editor
Allen Avner
University of Illinois

Newsletter Editor
Tom Reeves
University of Georgia

Special Interest Group Council Chair
W. Kirk Richardson
Georgia State University

Special Interest Group Representative
Peter Johnston
University of Alberta

ADCIS Executive Director
Gordon Hayes
ADCIS International Headquarters
Ex-officio

SPECIAL INTEREST GROUP COUNCIL

Computer-Based Training
John Buck
Technassociates

Elementary / Secondary / Junior College
Chris Dimas
Kennedy King College

Educators for the Handicapped
Susan Rice
University of Minnesota

Health Education
Arthur D. King
University of Florida

Home Economics
Elaine Muller
Denville, New Jersey

Implementation
Chuck Schneebeck
Fullerton College

Interactive Instructional System Presentation and Authoring
Nancy Marsh
Government Employees Insurance

Interactive Video / Audio
Bill Aggen
Metamedia

Management Issues
Sue Leslie
IBM

Mini / Micro
David Merrill
University of Southern California

Music
Charles Boody
Hopkins Public Schools

PILOT
W. Kirk Richardson
Georgia State University

PLATO
Peter Johnston
University of Alberta

Theory and Research
Mary Stoddard
Los Alamos National Laboratory

TABLE OF CONTENTS

COMPUTER BASED TRAINING (CBT)

CASIP - A Novel Authoring Tool for Open-Ended Language CAI . . . 1
Michael Anbar
Two Heads are Better Than One. Multiple Displays in Computer Based Training . . . 5
Tom D. Conkright, Scott R. Lucado
Computer-Based Army Job Skills Education . . . 8
Beatrice J. Farr, Sally A. Bell, Mark A. Sabol, Lois S. Wilson
Increasing CBT Developer Productivity with an Instructional Database . . . 11
Jesse M. Heines, Larry Israelite
Job Aids: Give the Readers What They Want . . . 15
Patricia Lawson
Design and Research on The Interactive Encyclopedia System (TIES) . . . 19
Janis Morariu, Ben Shneiderman
Tools for Managing CBT Courseware Development . . . 22
Timothy W. Spannaus

ELEMENTARY, SECONDARY, JUNIOR COLLEGE EDUCATORS (ELSECJC)

Relations: The Missing Link in Models of Cognitive Learning . . . 28
Gerald J. Blumenfeld
Implementing an Instructional Management System in Three K-12 Schools: A Project Report . . . 33
Michael L. Burger
Seven Local Area Networks . . . 36
Jane A. Malloy
Drills vs. Games — Any Differences: A Pilot Study . . . 40
David W. McMullen
Logo Tools for Teaching Algebra . . . 46
Nancy Roberts
Effective Administration of the Microcomputer Lab in Higher Education . . . 53
Charlotte Scherer
Gender Similarities and Differences in Sixth Grader's Attitudes Towards Computers: An Exploratory Study . . . 59
Marc Swadener, Michael Hannafin
The Relationship Between Students' Prior Programming Experience and Their Performance in an Introductory Pascal Programming Course . . . 66
Michael L. Waugh

HEALTH (HESIG)

Nutrition and Metabolism for Nurses: Basic Concepts in Nutrition . . . 72
George Kalnitsky
Is It Worth It? Determining the Cost-Effectiveness of Using Computer-Assisted Instruction in Nursing Education . . . 76
Donna Larson
Computers in Nursing: An Annotated Bibliography, 1979-1984 Using ZyIndex . . . 81
Barbara Thomas, Steve Bowers

HOME ECONOMICS CONSORTIUM (HOMEC)

A Research-Based Statistical Matrix for Demonstrating Impact of Computer-Based Learning . . . 84
James L. Morrison

INTERAGENCY GROUP FOR COMPUTER BASED TRAINING (IGCBT)

Using the Computer to Assist in the Management of a CBT Project . . . 92
Adele S. Suchinsky, David R. Losleben

IMPLEMENTATION (ISIG)

Institute for Computer Assisted Learning: An Alternative Approach to Implementing CAL in a University . . . 95
W. Bruce Clark

TABLE OF CONTENTS *(cont.)*

Interface Between Authoring Systems and Programming Languages - User-Defined Command 100
Yasuhiro Morimoto, Hiroshi Kato, Hiroyasu Chimura, Masahiko Kurata, Takahiro Sato
Implementation and Evaluation of an Original Psychology Software Series 105
Robert S. Slotnick, Abraham M. Jeger
KAPLEX: A Software Architecture for Knowledge-Based Authoring Systems 111
Y. Richard Wang, Jay F. Nunamaker, Minder Chen

MINI / MICRO COMPUTER USERS (MINI / MICRO)

Spreadsheets and Databases - Alternatives to Programming 116
M. Gene Bailey
The Administrative Evaluation of Microcomputer-Based Instructional Materials Development: A Case Study 122
George H. Conklin
A Nationally Validated, Exemplary Computer Management System 125
Debra J. Glowinski, E. Wayne Roberson
Computer Literacy - Myths and Delusions 128
Tommas J. Koehler
Computer Assisted Course Development and Instructional System (CACDIS) 135
A. L. Lakshminarasimhan
Streamlining the Software Development Process Through Design and Coding Guidelines 142
Mary Jac Reed
Distributed UNIX-Based CBT Authoring Networks 146
Charles V. Welsh, Thomas T. Chen

MANAGEMENT ISSUES (MISIG)

A Regional Standard for Developing and Documenting Spreadsheet Templates Within the Cooperative Extension Service 149
Margaret P. Ezell
Defining Intellectual Property Rights in the Distribution of Software 155
Margret Hazen, Thomas Lee Hazen
Protecting Your Development Investment: The Project Schedule as a Means to Control Risk 159
S. M. Leslie
Managing a Courseware Development Manager: A Primer 165
Valerie E. Mock
Aptitudes and Attitudes About Computers: Two States Compared 168
Paul J. Will

MUSIC EDUCATORS CONSORTIUM (MUSIC)

Structured Music CAI Programming and Data Structures 174
Ann K. Blombach
Computer-Based Recognition of Perceptual Patterns and Learning Styles in Melodic Dictation Exercises 179
Fred T. Hofstetter

PILOT USERS GROUP (PILOT)

PILOT as a User Interface to a Personal File System 187
John A. Starkweather

PLATO USERS GROUP (PUG)

The Evaluation of an Automated Test Delivery System (CMI) for the United States Army Engineer School 201
Tracie Carter
Efficient CAI in Engineering 208
Helen Kuznetsov
A Strategy for Simulations of Dispute Resolution Process for Use in Legal Education 214
Michael A. K. Lambiris
Academic Computing Philosophy: An Often Ignored Necessity 221
William W. Schnippert

TABLE OF CONTENTS *(cont.)*

Integration of Computer-Based Instruction Into Edmonton Police Department Training Program: A Case Study in Productivity 225
William A. Wosar, Michael Szabo

INTERACTIVE VIDEO-AUDIO (SIGIVA)

The Development of a Total Curriculum for Physical Science Using Interactive Videodisc Technology 231
Stephen F. Brown
Our Videotex is O.K., How's Yours? 236
Elisabeth M. Craig, Phil Gross
The Effectiveness of Interactive Video in Teaching Information and Applications 241
David W. Dalton
Quality Control in the Development of Courseware Utilizing Computer-Based Interactive Video 245
Richard A. Dassance
The Effects of Locus of Instructional Control and Practice on Learning from Interactive Video 242
Michael J. Hannafin, Maryanne E. Colamaio
Integrating Multi-Media CBI Approaches Into Simulator Training Programs 256
Michael J. Olson, Harold E. Popplewell
The Technology of Videodisc: A Forecast from 1986-2001 264
G. Steven Tuthill
Physics Experiments on Interactive Video Discs 503
Ronald Gautreau, Harpreet Chawla, Jenhaw Chang, Jihn-Fa Jan, Shuenn-Tyan Lee, William Savin

THEORY AND RESEARCH (SIGTAR)

Using Computer Displayed Pictures to Enhance Visuo-Spatial Aptitude in Pre-Adolescent Females: Suggestions and Opportunities for Research 269
Patti R. Baker, John C. Belland
The Role of Computer-Assisted Instruction in Mastery Learning 275
David W. Dalton
A Microcomputer System for Diagnostic Testing of Cognitive Errors 279
John M. Eddins
Using Technology to Measure the Effectiveness of Technology: Computers and Video as Data Collection Tools for Educational Research 283
Francis A. Harvey
The Effectiveness of Different Responding Approaches in Multiple-Choice of CAI Design 288
Jon-Chao Hong
Strategies for Performance and Learning with an Interactive Computer System 292
Elaine Lewis, Victor Shtern
Comparison of Foreign Language Learning with Computer-Assisted and Non-Computer-Assisted Study 296
Mary Ann Lyman
Relationships Among Knowledge Structures and Information in a Computer-Based Problem Solving Environment 301
Elise G. Millikin
The Effects of Textual and Animated Orienting Activities and Practice on Learning from Computer-Based Instruction 308
Lloyd P. Rieber, Michael J. Hannafin
Participative Multi-Role Instruction in a Distributed Knowledge-Base Environment 314
Taracad R. Sivasankaran, Ralph Ruby
Cognitive Error Diagnosis and Prescription for Fraction Arithmetic 324
Doris Smith Shaw
A Computer-Based Knowledge Extraction Tool: A Step in the Development of a Cognitive Skills Tutor 332
Mary L. Stoddard, Joe D. Emerson, Richard Kern
A Cognitive Error Diagnostic Adaptive Testing System 338
Kikumi Tatsuoka
An Application of Rule Space Methodology to the Assessment of Tonal Memory 343
Walter Vispoel

TABLE OF CONTENTS *(cont.)*

Cross-Cultural Replication: The Use of Concrete and Computer Simulation Activities for Learning Elementary School Mathematics Concepts ... 347
Arthur L. White, Donna F. Berlin, Pamela F. Adber

200 WORD ABSTRACTS (No Formal Papers)

COMPUTER BASED TRAINING (CBT)

Collaboration in the Design and Teaching of Computer Continuing Nursing Education ... 352
Carol A. Bauer
Computer Assistance in Foreign Language Instruction ... 353
Howard Jeffrey Bender
The Tri-Level Approach to CBT ... 354
Norm Carr, Geoff Burston, Michael Szabo
Making Intelligent Choices Among Microcomputer-Based Authoring Systems ... 355
Victoria L. Clegg, Jane M. Lommel
Production System Based Computer-Assisted Instruction ... 356
S. C. Davis, D. Sullivan, M. West
Machines That Teach: A History of Computer-Based Instruction ... 357
Brian L. Dear
Screen Design: Creative Uses of Simple Line Monochrome Graphics ... 358
J. D. Ellertson
A Program to Reduce On-the-Job Training for Air Traffic Controllers Through Computer-Based Instruction ... 359
Doug Farrow
DOE Contractor Radiation Safety CBT Course ... 360
Patrick R. Gardner
CBT Implementation: A Risky Business ... 361
Lori Gillespie
Instructional Technology Within the Supercomputing Environment: Using CBT as the Instruction Medium ... 362
Stephanie S. Goodwin
Have You Had Your Training Program Evaluated Lately? ... 363
B. H. Handler
Managing the Courseware Development Process ... 364
Ruth M. Hawkins
Teaching a Problem Solving Strategy Via Computer Introducing "EEV" ... 365
Bonney Hettinger
Computer-Based Training for Child Protective Service ... 366
A. N. Hewitt, C. LaBrosse, N. A. O'Neill
Design of a Computer Applications Survey Course ... 367
Norman Higgins
Computer Simulated Logic Circuit Instruction ... 368
Jon-Chao Hong
Embedded Training: The Explosive Growth Area in CBT ... 369
Julie E. Horine, Jeffrey B. Erickson
Discrepency Evaluation Model for Computer-Based Education ... 370
David L. Jelden
Direct Manipulation of Lesson Flow Structure ... 371
Robert L. Jorczak
Separating Course Logic and Course Content Development ... 372
Steven Just
Explorations in the Evaluation of Authoring System Software ... 373
Craig Locatis, Victor Carr
Courseware Development for the Insurance Industry ... 374
John A. Merrill
Streamlining Graphic Production Through Reusability ... 375
Mary G. Miller

TABLE OF CONTENTS *(cont.)*

Computer Assisted Creation and Analysis of Populations and Samples 376
C. N. Moore
Evaluating the CBT Course: Questionnaires Through CMI 377
Richard Reinhart
The Project Life Cycle for Developing CBT 378
Jerry Richard
The Initial Development of Subject Content for Instruction by Computer Technology 379
Grosvenor C. Rust
MIDIS - A Micro-Computer Based Flight Decision-Training System 380
A. Stokes, C. Wickens, T. Davis
Airborne Radio-Communication Training: Spatial or Verbal Problem? 381
A. Stokes, T. Davis, M. Weller, C. Wickens, B. Entwistle, R. Braune, A. Mane
CBT Project Decision Guide 382
Adele Suchinsky, David R. Losleben
Writing Effective Questions for Training Data Entry Operators 383
William Sweeters
Factors to be Considered in Expanding an Existing CBT System - Short and Long Range Planning 384
Barbara J. Wright

ELEMENTARY, SECONDARY, JUNIOR COLLEGE EDUCATORS (ELSECJC)

Computer Competency for Secondary Teachers 385
Lane F. Birkel
Concept Presentation Generator 386
Gerald Blumenfeld
Will Technology Improve Education in the United States? 387
Alfred Bork
Computerizing the Community College Campus 388
Linda A. Boyer
Computer Graphics for Kids 389
Barbara W. Fredette, Grace W. Ferrero
The Effects of Learner Control, Advisement and Prior Knowledge on Achievement and Instructional Event Selection in CBI 390
Jane Marie Gleason, Wallace Hannum
Three Tiered Approach to Institutional Computing 391
Floyd M. Grimm III, Richard V. Miller
Designing Effective Learner-Courseware Interactions: Research Based Recommended Actions 392
Wallace H. Hannum, David H. Jonassen
Progress Report: CAI for PASCAL in The Huntington Project Tradition 393
Jesse M. Heines
Science Instruction on Global Electronic Networks 394
James A. Levin, Michael Waugh
LOGO: A Language for All Grades 395
Robert A. Lucking, Cherie Harrell
Computerized Instruction and Distance Education: Penn State's Innovative Solution for the Isolated Adult Enrolled in Introduction to Statistics in Educational Research 396
Marilynne W. Mathias
Effective Teaching in Classroom Groups: A Computer-Based Alternative 397
Steven L. Robinson
Bibbits and Gribbits: An Adventure in Logical Thinking and Problem Solving with Primary Children 398
Charlotte Scherer
PHOENIX as a Cost Effective and Flexible Mainframe CAI System at the Junior College Level 399
Gary Vaughn
The Effect of Student Feedback on Achievement 400
Elizabeth West

TABLE OF CONTENTS *(cont.)*

HEALTH (HESIG)

Integration of Computer Assisted Instruction Within a Baccalaureate Nursing Curriculum 401
Jean Arnold, Elsie E. Gulick
Interactive Video Discs as Tutors for Visual Learning in Medical Education 402
Terry S. Field
Development of Interactive Videodisc Programs for Basic Medical Science Education 403
Ronald P. Jensh, F. Scott Beadenkopf
Developing CBE to Replace a Deceased Program 404
Victoria Moore
Productive Gaming 405
Victoria Moore
Routine Health Screening 406
Daniel K. O'Neil, Alan G. Adler
Building Expert Systems: A Powerful Instruction Technique 407
Stanley R. Trollip, Renate C. Lippert
The Development and Validation of Attitude Scales as Covariates in CBI Research 408
Charles S. White
Evaluation of Diagnostic Tests 409
Scott Worman
Using a Simulation to Teach Psycho-Socio-Economic Aspects of Clinical Medicine 410
D. P. Yens, J. T. Peters, M. C. Fahs, J. Lowe

HOME ECONOMICS CONSORTIUM (HOMEC)

The Computer as a Tutor in Dietetic Education 411
Leta P. Aljadir
The Effective Classroom Use of Microcomputers in Merchandise Planning 412
Holly E. Bastow-Shoop
An Interdisciplinary Approach to Utilizing Computers in Home Economics 413
J. Cheyer, B. Quinby
Know Your Man-Made Fibers 414
Nadine Hackler, Linda Heaton
Integration of CAD/CAM and the Interior Design Program 415
Susan Harshbarger
Introduction of Telecommunications into Vocational Home Economics Programs 416
Ann B. Kapp
Development of a Time Management Computer Program 417
Judith S. Kline, Emily S. Wiggins
Spread Sheets Facilitate the Teaching of Fashion Merchandising 418
Grovalynn Sisler
Shopping for a Refrigerator Via the Computer 419
Louise Wesswick, Edith A. Miller

IMPLEMENTATION (ISIG)

The Computer's Contribution to Solving Problems in Second Language Reading Tasks 420
D. Boatman, C. Cieri, N. O'Connor
Using Telecommunications to Encourage Reading and Writing 421
William H. Booz
The Equity in Technology Project: Bringing the Benefits of the Computer to All Students 422
Alice Fredman
Scholarly Webs: An Alternative Educational Software Paradigm 423
L. Nancy Garrett, Karen E. Smith
Applying Expert Systems to CBT 424
Greg Kearlsey
Method for Determining the Applicability of CAI, Interactive Video, and Teletraining for Delivery of Industrial Training 425
Mable B. Kinzie-Berdel

TABLE OF CONTENTS *(cont.)*

The Design, Development and Management of a Microcomputer Users Center 426
Randal A. Lemke
Evaluation of Videodisc-Based Training Technologies in an Armor Military Setting 427
Carl W. Lickteig, Billy L. Burnside
Strategic Plan for Managing Technology in the Workplace 428
Shirley Jean McCann
Distance Learning for In-Service Teachers Via Micro and Modem 429
D. McGrath, D. DePucchio, C. Franklin, D. Russell, C. Wilson, A. Hayes, S. Kane
An Integrated Approach for Implementing and Evaluating CBI in a Military Environment 430
T. M. Shlechter, D. M. Kristiansen, B. L. Burnside
Computer-Based Instructional Systems Utilizing Artificial Intelligence 431
Carroll L. Wilson

MINI / MICRO COMPUTER USERS (MINI / MICRO)

Evaluations of Educational Microcomputer Software: A Content Analysis of Reviews 432
Evelyn Bender
Computer Resources for Educators 433
Carol A. Dwyer
Large-Scale CBI with Micros: What's Missing? 434
Rick Hill, Leah Rampy
Simulating Complex Problems Using DYNAMO 435
Martin Nyhold
Full Circle Test Administration on Micros: Integrated Test Banking, Scoring and Grading 436
Judith Pine
Testing 1-2-3: Developing Benchmarks for Authoring Systems 437
Naomi O. Story

MANAGEMENT ISSUES (MISIG)

When to Use the Team Approach 438
Margret Hazen
Automating Cost/Benefits Analysis 439
Greg Kearsley
How Long Does it Take? A Detailed Log of a Software Development Project 440
Mary Jac Reed
Faculty Computing Education Needs: Results of the Ohio State University Study 441
Sallie J. Sherman, Ajay Bhardwaj

MUSIC EDUCATORS CONSORTIUM (MUSIC)

Contemporary Techniques for Contemporary Students: Computer Applications for Prospective Music Teachers ... 442
Judith Bowman
A District's Effort for Creative Use of the Computer in Music Classrooms 443
C. A. Brown, J. Crawford, J. Rehberg
Multi-Function Courseware for Music Instruction 444
Lan-Ku Chen, Jon-Chao Hong
Development of a Computer-Assisted Interactive Videodisc (CAIV) Program in Opera 445
Louise Cox
The Macintosh and CAIM: Developments, Potential, Problems 446
Warren E. Hultberg
The AMIGA and Music Applications 447
Philip L. McClure
Feasibility of Instruction with Interactive Videodiscs Created From Existing Video Media in Instrumental Music Education 448
Anne Miller
Exploring Powerful Ideas with Music LOGO 449
J. Kent Williams

TABLE OF CONTENTS *(cont.)*

PILOT USERS GROUP (PILOT)

Creating Instructional Interactions with PILOT Procedures 450
Gilberto Contento, Giuliana Cossi
The Use of Graphics in PILOT Programs 451
Lynda Doll
Incorporating PC/Storyboard Into PILOT Programs 452
Wing M. Fok, Kern Kwong, Howard Cheung
Transporting PILOT: A SUPERPILOT to PC PILOT Template 453
Sharon Gomez
Survival Spanish Interactive: A First-Year Interactive Video Course 454
Fred Jensen, Mary Ann Lyman
Acquiring Content Uniformity Across Multiple Sections of an Undergraduate Technical Writing Course: A Problem Solved with SUPERPILOT 455
Marilynne W. Mathias, Elizabeth Ellis
Microcomputer Diagnosis of Arithmetic Skills 456
Barbara Signer

PLATO USERS GROUP (PUG)

Experiences with a Small PLATO System 457
Iain Cooke
Students and Mainframe Authoring Systems: Stimulus for Development of CBI 458
Michael Neuman, Victor Showalter
Development of PLATO Usage Among Faculty 459
Deborah Ouelette
CBE: The Successful Implementation of PLATO in Public Schools 460
Pamela A. Wright

EDUCATORS OF THE HANDICAPPED (SIGHAN)

The Center for Special Education Technology 461
Susan E. Elting, Sharon L. Raimondi
Adaption of Tool Software to Assist Students with Special Needs 462
Sheldon D. Rosenberg
Visual Impairment and Computer Displays: The Effects of Foreground and Background Color 463
Michael John Simmers

INTERACTIVE VIDEO-AUDIO (SIGIVA)

Architecture for an Intelligent Tutor Using Interactive Video and Situational Diagnostics 464
A. E. Andrews, G. P. Lloyd, D. R. Powell, M. L. Stoddard
Interactive Video Courseware as a Research Tool 465
Janet White Azbell, Mary Lou Mosley, Naomi O. Story
The Merging of Videodisc and CD ROM Technologies 466
Greg Baur
Interactive Video in Today's Marketplace: Industrial Training, Management Training, and Point of Information 467
Brenda Bracket, Margaret A. Chopp
Design for a Computer-Interactive Videodisc Map Orientation Simulation 468
A. B. Cox, L. W. Carstensen, J. F. Moore
Creating Adaptive Learning Environments Using Existing CBIV Courseware 469
Russett Ann Craig
A Project to Enhance Training Videotapes with Computer-Supported Interactivity: Lessons Learned 470
Doug Farrow, Devah Galloway
The Development of an Interactive Videodisc Program to Deliver Pesticide Applicator Training in Virginia 471
Michelle M. Geasler, Michael J. Weaver
An Interactive Videodisc Program on Teenage Suicide and Depression 472
Carol Hargan

TABLE OF CONTENTS *(cont.)*

Computer-Based Interactive Videotape Simulations in Pre-Service Teacher Education Programs 473
Anne M. Hayes
Interactive Video Instruction and Learner Control 474
Michael C. Hetzel
Matching Video Tape and Interactive Video Disc Instructional Methods to Students Field Independent/Dependent Preference 476
Nancy P. Houston
Student Tour of a VAX 11/785 Via Interactive Video 477
Frederick J. Jenny
Compact Disc and Videodisc Technology: An Overview of Similarities and Differences 478
Carolyn S. Lay
A Comparative Review of Audio Methodologies in the CBIV Environment: New Design Capabilities 479
John F. Moore, Mary G. Miller
Getting Involved in Interactive Video: Problems and Pitfalls 480
Terry A. Olivier
Beyond Tape Versus Disc: Issues in the Design of Interactive Video 481
Timothy L. Phillips, Michael J. Hannafin
Overlay Graphics for Interactive Videodisc Systems 482
D. V. Pigford
IBM Infowindow System: A New World of Interaction and Excitement for Videodisc 483
Marguerite Raaen
A Bargain Basement Generic Interactive Videodisc 484
Mark Sanders
The Design and Evaluation of Interactive Video Disc Lesson in Chemistry 485
S. G. Smith, L. L. Jones, M. L. Waugh

THEORY AND RESEARCH (SIGTAR)

The Role of the Computer in Second Language Acquisition Research 486
Dana Boatman, Christopher Cieri, Nadine O'Connor
The Use of Expert Systems and Semantic Networks in the Explanation of Physics Principles to Students in an Intelligent CAI System 487
Stephen F. Brown
The Crafty Art of Using Analogies 488
Kathy A. Fletcher, Janis Morariu, Marcia Whitney
Relationships Between CBI and Research on Teaching 489
John Guthrie, Mark G. Gillingham
The Impact of Computer Usage on Understanding Mathematical Operations 490
W. Hannum, H. Ballew, J. M. Gleason, J. Kolb, C. Stewart
The Effectiveness of Different Responding Approaches in Multiple-Choice of CAI Design 491
Jon-Chao Hong
The Effects of Computer-Based Knowledge Representation Strategies in the Acquisition of Selected Microeconomics Concepts 492
Vincent E. Lasnik
The Application of a Model for Designing Interactive and Responsive Courseware 493
Cindy B. Leshin, Vernon S. Gerlach
LOGO Styles and Ability in Seventh Grade Students 494
Lucy B. Mallan
Report on UCLA LOGIC Software Project 495
Robert L. Martin
Qualitative Research: What Can it bring to CBE? 496
Delia Neuman
ICAI: Intelligent Application AI 497
Ok-choon Park, Robert J. Seidel
Adaptive External Pacing as an Alternative to Self-Pacing in CAI 498
Kyle Peck

TABLE OF CONTENTS *(cont.)*

The Effects of Practice and Orienting Activities on Learning in an Interactive Video Lesson 499
Timothy L. Phillips, Michael J. Hannafin, Steven D. Tripp
The Evolution of Computer-Based Instruction: Behavioral to Cognitive 500
Lloyd P. Rieber, Michael J. Hannafin
Models for the Development and Design of CAI: An Analysis of Comprehensiveness and Operational Level 501
Robin Taylor
Visual Metaphor as an Instructional Strategy 502
Steven D. Tripp

ADCIS Proceedings, November 1986

CASIP - A novel authoring tool for open-ended natural language CAI

Michael Anbar PhD

Department of Biophysical Sciences, School of Medicine, SUNY, Buffalo

ABSTRACT

CAI that uses open-ended questions, and calls for answers in natural language, is the preferred method of instruction in many training situations. A novel authoring tool has been developed to meet the specific needs of open-ended computerized instruction. This program, named CASIP, is essentially a single-purpose interpreter (written in C) which uses instructions contained in an ASCII file to interpret natural language in a prescribed context and to respond accordingly. This and other ASCII files, which contain the questions and the responses used during a learning session, are completely loaded into memory at the beginning of the session. The use of CASIP involves no preprocessing of the ASCII input files. Authoring with CASIP requires no programming knowledge and it can be done with any word processor or line editor. CASIP can be used to generate in a few days sophisticated open-ended instructional sessions on any subject.

Open-ended versus Multiple Choice CAI

The popular multiple choice question (MCQ) approach in instruction has serious limitations. Its major drawbacks are: (1) It is the instructor and not the student who formulates the alternative answers to a given problem. (2) It assumes that a finite small number of plausible answers (including credible detractors) constitute essentially all the reasonable alternative answers to a given problem. (3) It implies unambiguous problems. Except in mathematics or in trivial recall of facts, there may be many reasonable answers to a question, some of which may not have occured to the instructor. Still, the student is expected to select the "correct" one(s). (4) It does not allow the student to explain the reasoning behind a certain choice; students may come up with the "right" answer for a wrong reason, or with a "wrong" answer for a valid reason. Students who think creatively may thus be at a disadvantage in a MCQ environment.

These limitations are pertinent to most instructional situations, with the possible exception of trivial memorization and recall drilling. This is true of high school and undergraduate college teaching as it is in the teaching of medicine, law, economics or any other subject that requires conceptual problem solving. Further, in computerized MCQ students can get immediate feedback to their answers. However, such feedback just confirms or contradicts the answer and not critically analyse the rationale for the given choice. This is generally not done because it is very difficult, if not impossible, to apply MCQ's to examine the reasoning of the student without introducing biases - suggesting reasoning the student may never have thought of.

The alternative to MCQ is cue-free open-ended questions (OEQ). Here the students who are presented with a problem must come up with a plausible solution all on their own. Any teacher who has graded short assays or open-answer questions has realized how more informative they are about the student's knowledge and mode of reasoning, in comparison to MCQ. In fact, coming up with a set of alternative potential answers to a given problem is an experience that often is educationally more valuable and intellectually more rewarding than chosing the "right" alternative from among those answers. When used in an iterative manner OEQ are well suited for the age proven socratic interactive mode of instruction which is based on a teacher-student dialog that effectively explores and criticizes the reasoning of the student. Furthermore, the OEQ modality of CAI is a man-machine interactive experience where both parties use essentially the same language.

Programming of Open-ended CAI

The computer has proven to be highly efficient in handling MCQ, especially since it facilitates multiple stage conditional branching. However, to become a major component in advanced academic instruction, the computer must be capable of handling OEQ -it must be capable of understanding natural language. The handling of the whole universe of words and expressions in response to a question may seem freightening at first sight. However, based on first hand experience with this type of CAI for several years [Anbar, 1984,1985,1986a,1986b], I can state that the range of principal key-words (each with a number of, say, up to 10, logical synonymes) that may show up in a coherent answer following a well structured open ended question is rather limited (between 5 and 20). This relatively small number of string recognitions is well within the capacity of computers; any microcomputer for that manner. But it still means the precise recognition of over a hundred words or phrases (a phrase in this context is a coherent combination of two or more words) for each question, which is not trivial from the programming standpoint.

Using the classical programming approaches and high languages such as BASIC or Pascal it is far more difficult to program OEQ sessions than branched MCQ, and the time and effort required make it virtually impossible to generate instructional materials of acceptable quality in quantities that can might have an impact on the curriculum. Unfortunately we also cannot benefit from the use of commercially available "authoring languages". When applied to our specific goal of generating effective OEQ materials, non of the available software has mets our needs. For one, none of these packages can handle hundreds of recognizable words and tens of potential answers to each question and then branch accordingly into tens of branches, repeating the same diversity of response over and over again. Even if coded to do so by a *tour de force*, the response times obtained are prohibitively long (tens of seconds per cycle!). This is not surprising since these programs have not been designed to handle OEQ.

Special Requirements of Open-end CAI

The only viable solution to our problem has remained the generation *de novo* of an authoring tool that will meet our specific needs. These needs can be generically expressed in the following five requirements:

A. Since the recognition of whole sentences or parts thereof is utterly impractical without parcing into words, we need a program that will allow the rapid (within less than 2 seconds) matching of each word in a sentence of a student's answer (averaging, say, 8 words) with any of a few hundreds of key words at a time. The program must also be able, within the same time frame, to branch to any desireble response after recognizing 1 to 6 different key-words in a sentence. The recognized words may appear in a prescribed sequence and at a specified separation from each other (adjacent two words, or two words within, say, four words of each other). It must also readily allow alternative treatment of an answer upon recognition of the presence of certain words (eg. "not" or "never") that might change the meaning of a sentence.

B. The response to a recognized word or sequence of words or to the absence of a word or sequence of words in the student's input, may be a short sentence, with a follow-on question (that specifies *its* expected range of responses), or it may be a display of text or tabulated numeric data (ranging from several lines to several sceens), computer stationary graphics (binary stored pictures), *in-situ* generated graphics or computer animation, followed by the next question. The program must also recognize a repeated answer that might prompt the same response (to avoid logical loops). The program must allow to make the response conditional on answering a prescribed previous question (or number of questions); this requirement is essential when analysing the reasoning of a student. It must also provide a different response after the student has given a prescribed number of unexpected answers to a given question (to avoid the student's frustration). Finally, it must allow the incorporation of a preselected key word from the student's answer into the response or the follow on question; this gives the student the assurance that the computer really "understands".

C. The program must produce a permanent record of the student's answer (verbatim), of the question that evoked that answer and of the response the machine gave following the given answer. This is essential for the development of any OEQ material. Having a record of the unexpected answers of several students in trial runs is absolutely necessary to fine tune the program before its deployment to the whole class; the author alone cannot anticipate all possible answers to each question.

D. The program must have the capability to display randomly selected questions or responses from a specified series, in situations where virtually the same response is called for (eg. responding to an unexpected answer or to a repeated answer), to make the machine look less predictable and, therefore, less mechanistic. This is especially important when the same material is to be used several times by the same student.

E. The input to the program - questions, responses, data, as well as the correlation between key-words recognized in the student's answers and the corresponding responses, must be presented in a simple format, so that an author without any programming experience should be able to write lessons with minimal difficulty. The author should have to do minimal manipulations when incorporating the authored inputs into a working program (preprocessing or compilation should be avoided). This is especially critical during development of a complex OEQ session and especially when additions and corrections are repeatedly made following trials with students.

Considering this combination of requirements, it is clear why it was not met by any of the available commercial authoring languages; the developers of the latter did not have these objectives in mind. What we have done is to use the C language, which combines high flexibility with speed of processing, to develop a programming tool that meets all these goals and more.

CASIP - an Interpreter for Open-ended CAI

We have called our OEQ program CASIP (Computer Assisted Socratic Instruction Program) since it facilitates computer-mediated socratic teaching. CASIP has been written in the C language in a fairly compact manner (<40K compiled, with a source code of about 30K). It runs on a 192K IBM PC (with DOS 2.0 or 2.1, or 256K with DOS 3.0 or 3.1) and is expected to be transportable to any MSDOS 8086/8088 system. It can run on a mono monitor (without graphics), on a color monitor, or on a dual mono/color system (ideally). Since the graphics are external to the core program, any compiled graphics (or music, if desired) in an .EXE file, or binary coded screens (16K) in a .PIC file, can be displayed as part of the response.

All the author's inputs - the questions, lists of recognizing words, responses and the correlation table, which provides the run-time correspondence between the recognized key-words and the responses, are simple ASCII lists (they can be produced on any word processor or line editor). All these files are completely loaded into memory before the program starts. This facilitates a high speed of processing of the special algorithms that minimize the time of answer analysis. These ASCII inputs may amount to as much as 100K combined, but as little as 20K input (about 5 typed pages) can produce a very effective instructional unit.

The program reads external text files, it rapidly reads binary graphics files and executes external graphics .EXE files (in compiled BASIC, Pascal or C). It should be emphasized that the use of CASIP requires no preprocessing or compilation. In fact, this program acts as a single-purpose interpreter which immediately runs the loaded ASCII files. It records the student's answers fully, keeps a record of the preceeding question and the follow-on reponse, and keeps a record of the time (with a precision of seconds) each answer was given; this helps to identify difficult questions or casual, not sufficiently thoughtful answers. The record includes also the number of times the student gave an unexpected answer and the score; each response or question, if you wish, may have a score assigned by the author. The student's record is eventually written into an individualized hidden (optional, activated by external control) file.

The program can also be instructed to terminate at a predetermined time between 10 and 90 minutes (set at 10 minute intervals), or after a predetermined number of question-answer cycles. It can be also terminated as one of the responses following a certain answer - termination on accomplishment of a prescribed learning task. A special instruction traps control-C termination and warm booting to avoid unwarranted termination by a frustrated user. The instructor can, however, terminate the session at any time using a special secret (known only to the instructor) key-word. Another instruction calls for a continuous clock display on the screen.

The options of word recognition include the recognition of a single or a sequence of words (usually 2 to 6 words); words in a sequence do not have to be adjacent to each other and the author may specify how many other words can be accepted between each word pair in a sequence. The author has also the option to allow two words to be recognized irrespecive of the order they appear in the answer. The author can also specify one response to a combination of two words (in sequence or, optionally, out of sequence), another response if only the first word was recognized, and yet another response if the second word alone was recognized. As mentioned above, there is the option to modify the response even when the desireable sequence of key-words was recognized, if a certain other word (eg. "no" or "never") were recognized in the answer. Anther option is to display automatically a certain response after the student answered satisfacorily a certain number (say, four) out of the possible answers (there may be, say, 12 different answers to a given question the author anticipated) to a given question. Automatic response may also be given following a sequence of (say, three) unrecognized (wrong or unexpected) answers.

CASIP in Use

CASIP gives the author all the necessary flexibility to recognize the contents of a student's answer at any desireable level of precision. Obviously, the more restrictive the prescription of the expected answer the more answers must be prescribed to a given question or the greater are the chances to receive an unexpected answer. In other words, the author must learn to compromize between the chance of accepting a poorly matching answer (a false positive, in the medical lingo) and unjustifiably rejecting an acceptable answer (a false negative). Trial runs with students is the only way to minimize authoring errors of either kind. The ease of obtaining a record of the students' answers and the ease of correcting the input files make this iterative process very easy. It takes just an hour or two to make substantial revisions in the files following a productive trial run.

The authoring process in generating a CASIP session has essentially nothing to do with computer programming. What the author has to do is to carry out an imaginary dialog with a student and type the questions, the keyword-words to be recognized in the "student's" answers, and the responses, in a conventional word processor file. The rest includes separating the three types of inputs into three different files (by block movement and transfer), and the construction of the correlation table that instructs CASIP which response will follow a given set of recognized key-words, which question will follow that response and what group of key-words to look for in the answer to that question. Because of these minimal requirements, the authoring of a sophisticated session takes only two to three days. Including the repeated testing on students, a session can be ready for class use in a week or two.

We have tested CASIP on a variety of subjects - from emergency medicine to social studies at the high school level, and the results are highly encouraging. We have thus demonstrated that effective open ended CAI, which uses unrestricted natural language as its input and can be applied to any academic discipline at any level of education, can be produced with an unexpectedly low level of effort.

[1] Anbar M., (1984) Experiences with CAI in Medical Education. in Capitol-izing on Computers in Education, Proc. 1984 AEDS Convention, C.D. Martin and R.S. Heller, Editors, Computer Science Press, Rockville, 1984, pp 188-90.

[2] Anbar R.D., Anbar M. and J.M. Hassett Jr, (1985) Emergency Medicine and Intensive Care Simulators. Proc. 9th Annual Symposium on Computer Applications in Medical Care (SCAMC), Ackerman M., Editor, IEEE Computer Soc, pp 547-550.

[3] Anbar M., (1986a) Open-ended Computer Based Medical Instruction - Avoiding the Pitfalls of Multiple Choice Behavior. Resident & Staff Physician, August 1986, in print.

[4] Anbar M., (1986b) The Socratic Teacher - Computer Assisted Self Instruction. Ch 9 in: Computers in Medicine, M. Anbar, Editor, Computer Science Press, Rockville, 1986.

TWO HEADS ARE BETTER THAN ONE!

Multiple Displays in Computer Based Training

Tom D. Conkright Rediffusion Simulation, Inc.

Scott R. Lucado Rediffusion Simulation, Inc.

From the perspective of the user of computer-based instruction or interactive video, the most significant piece of equipment is the display screen. This is the point where the learner's eyes are focused nearly all of the time. Therefore, the quality and completeness of the display interaction can be critical to a program's success.

Resolution or the number of picture elements (pixels) is a major contributor to determining the amount of information and the quality of the presentation. However, even with high resolution displays, it is frequently difficult to have all of the pertinent information showing at one time. Instructional effectiveness demands that the learner be able to visualize a total setting, procedure or operation. In many cases, it is useful or even necessary to have more than one display in order to meet this instructional demand.

There are some people who insist on single displays. Single displays are simpler to use and simpler to develop materials for. Proponents, however do not cite research or learning theory that favors a single display over multiple displays.

One frequently expressed supporting argument for a single screen display is the mentality of movie goers and television watchers. While most of these presentations are indeed single screen, all of us are familiar with the very sophisticated presentations that have been made with multiple movie projections in such settings such as Omnimax theaters or the Epcot Center. And of course CBT is not casual entertainment and should not be hampered by entertainment media.

In the earliest development of interactive video, technology limitations demanded the video be presented on a separate screen from the computer images. Newer technology boards, to overlay video and computer images, are now commonplace so that some of the technical need for two displays has decreased. However, overlaying is primarily done today in lower resolution systems with interlacing of both images. As the CAI community continues to recognize the significance of higher resolution and non-interlaced displays, this issue keeps coming back.

For an example, the OnLine board supports a 512 x 480 pixel resolution with 4096 colors. In order to acheive this added performance from a standard micro computer, DOS generated text must be on a separate display. This is especially significant during authoring.

In a printed environment, all of us at one time or another have used multiple points of attention--that is to say, more than one piece of paper. For example, in preparing for an exam or writing a report, it is most common to have more than one reference open at a time.

In the computer environment, it is common to support screen activity with printed text materials. In addition, on the screen, we now have well over two years of emphasis on windowing. Windowing produces multiple displays in one viewing area, but has the same effect as multiple displays. The limitation is that the window area may be too small or may obliterate or interfere with the primary information displayed on the screen.

Cockpits, power station controls and many other technical settings have always demanded that the operator be able to monitor separate indicators and respond to any of them rapidly. A pilot who cannot keep track of material on two or three separate computer displays in a training environment probably has little place in the cockpit.

Overall, in real life, in other training media and in other computer applications we find a wide variety of examples where the user is expected to keep track of multiple points of interest. So why not in CBT?

Acknowledgement: We are indebted to our colleague David Welham, at Rediffusion Simulation, Ltd. in the United Kingdom for specific details and understandings in this paper.

Multiple Display Examples

There are already many CBT projects that are using multiple diaplays. The following are a few examples that illustrate the need and usefulness:

1. Bell Helicopter Textron, OH-58D (U.S. Army)

Bell has been involved for the past three years in a major conversion of the helicopter that is the observation platform for the U.S. Army. This conversion affects nearly 90% of the airplane, installing some of the most advanced communications, avionics and flight systems available. This has a significant impact on the training of both members of the flight crew and the full range of maintenance personnel.

The solution configuration is two screens; one with video and computer graphics overlay, and one with only the computer graphics. Both screens are equipped with touch panels. The displays are supplemented with compressed audio on the video disks.

A principal reason for selsction of this particular two-headed approach was to simulate the helicopter's Multi-Function display screen for both crew members. Using two monitors permits effective training for flight crew members, either individually or in pairs.

2. Federal Express Flight Training (B727 & DC10)

Federal Express has one of the largest fleets of aircraft in the U.S. and a demanding flight schedule that is unequalled in the industry.

The configuration here is similar to Bell's except the video resolution has been increased from 480 lines interlaced to 512 lines noninterlaced and the compressed audio has been replaced by digital audio. The courseware design in both cases is similar; menus, instructions, feedback, concept graphics and detailed control panel graphics are presented on the computer only display. Detailed video, both still frame and motion, is presented on the second screen with overlaid box, circle or pointer highlights and text labels. It is possible, for example, to display a simulated throttle and stick on one screen and horizontal and altitude indicators on another.

3. Central Electricity Generating Board (U.K.)

The power industry requires a highly complex training regimen to prepare engineers to keep a steady flow of electricity regardless of power demands and system breakdowns.

CEGB selected three* displays for maximum flexibility. The far left hand display presents video, the center display provides high resolution computer graphics with touch panel, and the right hand display provides additional color graphics. Power system control technicians and engineers can view, on the video, gauge and indicator reactions to normal and fault conditions in a real control room, and the arcing of electricity in real power lines. From the touch panel display they can control power and exitation in order to reestablish system stability. In doing so, they can call upon a library of swing curves and vector diagrams that are being updated by the computer and can be displayed on the graphics screen.

4. Canned Foof Sterilization Process (U.K.)

In the manufacturing world, control panels are obvious and lights, switches and readouts readily visible. These present interesting opportunities for CBT, but of equal challenge is the need to present a conceptual understanding of processes that are not visible.

A two screen arrangement for a company in the U.K. allows the student to generate various control panels on one screen to see the effects upon a schematic representation of parts of the sterilizer on the other. Over 20 separate graphics are linked to a complex math model of the process. Any graphic can be displayed on either screen of which one is touch sensitive. A menu of fault conditions allows the student to produce realistic troubleshooting in the protected environment facilitating direct learning to the real task.

5. Singapore Airlines Flight Training (B747)

In this configuration two terminals are interfaced to a mini-computer containing the data bases and math models of a full flight simulator. A CAI terminal with touch panel provides realistic graphics of control panels, such as the fuel system. The user interacts with this terminal which provides the input to the simulator computer. The output is a schematic representation on a second high resolution graphics monitor. As the learner manipulates an aircraft subsystem, such as hydraulics of fuel, he can see the results immediately on the schematic.

6. Other Potential Examples

There are many other potential applications in government and industry. For example, in a medical emergency simulation, one screen can be dedicated to furnishing patient status while the medical practicioner interacts with the second screen to "save" the patient.

Another example concerns teaching computer applications. A manager learning a complex inventory system, could keep and manipulate a full standard screen of supplies and have a second screen that is the tutor and tester.

---

* Three heads are better than two!

Development Issues

Two or more screens adds to the complexity of development. Decisions need to be made about:

--what information is best presented on which screen,
--where should the learner interact,
--how much changes when there is learner input.

Since most of these multiple displays combine both video and computer images, there are decisions about what should be presented on which medium. Audio is yet another set of decisions. Making all these decisions and keeping track of the branching, flow and calculations adds considerably to the task of designing and checking out courseware.

A rule-of-thumb estimate is that the design effort is approximately thirty percent greater than a similar program using only one screen.

Programming will also require more effort. Although the basics of programming do not change, the added control of an increased number of peripheral devices (often including not only a second monitor, but a second touch panel and/or an additional audio device) can be a taxing problem. The range may be an increase from as little as five to as much as fifty percent. The key to control is very careful planning in advance. The programming is rigorous and requires a person who is comfortable with several things happening simultaneously. This is usually not a good introductory project for a new developer.

Additionally, the quality control process is legnthened. As the programs' complexity increases, the possibility of something going wrong increases greatly. Reviewers need detailed checklists to assure that everything is working when and how it was planned.

Summary

For computer-based training to realize its fullest potential, the instructional design should not be hampered by artificial equipment configuration limits. We would probably see more examples of multiple screens if the media selection process included this potential. Multiple displays are often needed to provide for a wholistic presentation and realism in the learning setting. In other cases, a second display is needed to prevent interference occuring on the primary diaplay. Although there are obvious increased costs associated with the equipment and development effort, those who have successfully implemented multiple displays have concluded that the results justify the investment.

Contact:
Tom D. Conkright
Rediffusion Simulation Inc.
2200 Arlington Downs Road
Arlington, TX 76011

817-640-5000

ADCIS Proceedings, November 1986

Computer-based Army Job Skills Education

Beatrice J. Farr, Sally A. Bell, and Mark A. Sabol

U.S. Army Research Institute

Lois S. Wilson

Hazeltine Corporation

ABSTRACT

The Job Skills Education Program (JSEP) is a computer-based curriculum which will eventually replace the Army's current Basic Skills Education Program. It will provide standardized instruction (on both PLATO and MicroTICCIT) for soldiers who demonstrate deficiencies in the knowledge and skills needed to learn and perform their Military Occupational Specialties (MOS). JSEP is based upon an exceptionally detailed analysis of the tasks performed in the most populous Army jobs and of those skills all soldiers need to know. Although it includes basic math and verbal lessons, JSEP is not a traditional basic skills program. The program's uniqueness lies in its offering of such non-typical competencies as: identifying the symbols on a flow chart, using hand and arm signals, and minimizing safety and security risks. Tryouts are currently under way at four Army sites; third party evaluation will be performed once the program is in place. The sponsors of JSEP are now consulting with federal and state agencies to determine effective technology transfer strategies. Initial reactions to JSEP suggest that it will have applicability in efforts to improve the employability of youth and the re-employment of displaced adult workers.

No one questions the vital importance of training as a mechanism for assuring that the Army can accomplish its missions. The role of education is perhaps less clear cut, even though it dates back to the very beginning of the U.S. Army. We are a far cry from the problems of the days when itinerant ministers tutored illiterate Revolutionary War troops; however, the educational deficiencies of soldiers is still the subject of diverse and serious criticism (Dewitt, 1979; GAO, 1983).

The demands of a high-technology Army provided the impetus (beginning in the early 1980s) to revise, upgrade, and expand the offerings of the Army Continuing Education System (ACES). The Job Skills Education Program (JSEP) which is the focus of this paper, is one of several efforts aimed at improving Army basic skills instruction.

JSEP is a multi-phased program begun in fiscal year 1982. Currently, it is being developed by Florida State University and the Hazeltine Corporation and monitored by the U.S. Army Research Institute (ARI). The sponsor, the Education Division, Office of the Deputy Chief of Staff for Personnel, expects JSEP to replace the Army's current Basic Skills Education Program when it is implemented. JSEP will provide a standardized curriculum for soldiers who demonstrate deficiencies in the knowledge and skills required to successfully learn their Military Occupational Specialty (MOS). It is expected to enhance enlisted career potential by improving soldier job performance.

In accordance with current policy, JSEP will be an on-duty program. It will use a computer-based management system to facilitate open entry/open exit. At present, most of the lessons being developed will be computer delivered; however, the plan calls for using existing materials and incorporating materials developed as part of other ARI efforts, whenever appropriate.

A unique aspect of JSEP is that it builds upon a very detailed front-end analysis of MOS baseline academic skills related to job performance. This analysis covered tasks performed by soldiers in the 94 highest density MOS, in addition to common tasks (the skills that all soldiers, regardless of their MOS, need to know). Although the Army has over 300 MOS, the 94 covered in the analysis represent about 80% of all soldiers. Perhaps the most useful product developed as part of the analysis was a taxonomy listing more than 200 prerequisite competencies for these MOS. The competencies were derived from detailed reviews of soldier manuals, and from extensive interviews with subject-matter experts at Army schools. The baseline skills analysis also resulted in a series of tests intended to diagnose deficiencies in the prerequisite competencies. Items from these tests will be used in JSEP.

JSEP will include a front-end learning strategies module designed to improve soldier skills in reading, studying, test taking, and problem solving. The curriculum will consist of this strategies-training, plus 180 diagnostic review lessons and 120 skill development lessons which are being developed for the PLATO and MicroTICCIT computer systems. The program is being tested at four sites, prior to an Army-wide phased implementation.

To help put JSEP into perspective, the remainder of this paper will use a format that we have found useful in other contexts. It will pose, and answer, a series of questions relating to what JSEP is and is not, why it was undertaken, how it differs from other basic skills programs, and what features it has that make it likely to improve training and job performance.

Question: What were some of the most serious criticisms of earlier Army basic skills programs that JSEP will attempt to address?

Answer: Overall there was a failure to:

1. Provide instruction to eligible soldiers.
2. Standardize instruction.
3. Centralize management.
4. Utilize technology to improve education.
5. Identify skills needed for each MOS.
6. Assure that all on-duty instruction is job-related.
7. Relate instruction to job performance.
8. Implement appropriate test measures.
9. Evaluate program effectiveness.

Question: JSEP is called a job-related program. Is it the same as job training?

Answer: No, JSEP is education. It is very important to understand this distinction. JSEP does not teach the actual job skills that soldiers use in their MOS (e.g., repairing a tank, cleaning a weapon). Rather it teaches what we call prerequisite competencies. These academic competencies are the underlying skills (such as following directions from a diagram) that enable soldiers to get the information needed to repair that tank or clean that weapon.

Question: Most JSEP lessons are presented in a military job-related context. How does this differ from previous programs and how is it expected to help?

Answer: Earlier programs frequently relied on materials that were not developed for an adult population. Furthermore, the skills taught were often not related to job requirements.

While all of the skills in JSEP are job-related, very few of them are strictly military (e.g., identifying grid coordinates on a military map). Although virtually all lessons would be equally relevant in civilian jobs, virtually all are presented in a military context--in other words, they have been "greened". Research provides some evidence that individuals retain job-related skills best when the original learning context is also job related.

Question: How does the remedial nature of JSEP differ form other basic skills programs?

Answer: When speaking of JSEP, it is important to keep the objectives of the program in mind. The only soldiers who will receive JSEP will be those who have been diagnosed as deficient in academic skills needed in their MOS. So, in that sense it is remedial. But it is not remedial in the ways of typical basic skills curricula. For example, it is not aiming for any given grade level performance, nor does it attempt to teach non-literate soldiers to read. There are other Army programs to address these needs. What JSEP does do is teach a very broad range of underlying skills that are related to the high density MOS.

Question: What are some examples of job-related skills taught in JSEP that go beyond traditional curricula?

Answer:

1. Select appropriate decision or course of action in a specified situation.
2. Identify the meaning of symbols on a flow chart.
3. Use and interpret hand and arm signals.
4. Apply preventive measures to minimize potential safety and

security problems.

5. Write a report that justifies actions taken and provides good reasons for rejecting alternative actions.

Question: How does the lesson philosophy of JSEP differ from that of more traditional basic skills programs?

Answer: The assumption is that there may be two reasons for soldiers to fail items on tests of prerequisite competencies.

1. They never learned the material covered by the test questions, and may be lacking one or more subordinate skills related to the test items; or, 2. The soldier may have once known the material, but now cannot recall it.

The JSEP concept calls for giving soldiers short diagnostic review lessons related to their MOS. This may be sufficient to refresh their skills or knowledge; but, if they still cannot answer test items, they will be sent to longer skill development lessons which contain much more elaborate instruction and feedback.

Questions: What advantages are seen for using computers in JSEP?

Answer: The soldiers who will be the population for JSEP have, at least partially, failed to achieve what they need through traditional education. Computers offer a novel, interesting, motivating, and infinitely patient way to provide instruction. Moreover, computers for JSEP will permit:

1. Easier update and revision of lessons.
2. Less reliance on expensive, highly trained instructors.
3. More standardization of education throughout the Army.
4. Better management of soldiers' instructional programs.
5. More accurate evaluation of student performance.
6. More complete records of student progress.

Equally important, computers make it easy to regulate open-entry/open-exit, thereby facilitating individualized self-paced instruction.

Question: JSEP is being tried out at various Army sites, what are these tryouts supposed to accomplish?

Answer: They will provide information on: (1) Whether the lessons teach what they are supposed to teach and, (2) the adequacy of the soldier management system, the reporting system and the JSEP testing program. It should be noted that even with all this, the trials are not designed to tell whether JSEP works as a total system. The developers are testing the various components before turning over the completed product for third party evaluation.

Question: How will JSEP be evaluated to see if the total system really works and if soldier performance is affected:

Answer: There is no simple, short-term answer to this. We will know, after external evaluation, whether lessons teach, whether the instructor training is adequate, whether the management system and testing programs operate as they were designed to, etc. The bottom line, though, is whether or not JSEP has an effect on soldier job performance. This information, of necessity, will only emerge longitudinally, as soldiers in sufficient numbers go through the program. Gathering this type of data is never easy, and for JSEP the task is compounded. This is because the standardized tests used in other programs are not appropriate measures of success in JSEP. Furthermore, developing valid and reliable job-performance measures is notoriously difficult. Initially, therefore, success in JSEP will be judged in terms of surrogate measures of job performance. As the Army develops appropriate job-performance measures, they will be employed in the program.

REFERENCES

Dewitt, Sue. Report on the Army Continuing Education System. Memorandum for the Secretary of the Army. 1979, Washington, D.C.

General Accounting Office. Report to the Secretary of the Army, GAO/FPCD-83-19, 1983, Washington, DC

Contact:

Beatrice J. Farr
U.S. Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333-5600

202-274-5540

INCREASING CBT DEVELOPER PRODUCTIVITY WITH AN INSTRUCTIONAL DATABASE

Jesse M. Heines, Ed.D.

Asst. Prof. of Computer Science
University of Lowell
Dept. of Computer Science
One University Avenue
Lowell, MA 01854

Larry Israelite, Ph.D.

Director, Training Products
Scientific Systems, Inc.
One Alewife Place
35 CambridgePark Drive
Cambridge, MA 02140

Abstract

Computer-based training (CBT) course developer productivity is most often expressed as the number of development hours needed to produce one hour of on-line instruction. CBT authoring systems offer a number of approaches to increasing developer productivity, generally by providing built-in features for typical instructional sequences. As one would expect, each approach has its benefits and shortcomings. We have adopted an "instructional database" approach, which attempts to provide simplicity for the course developer without sacrificing true programming power. In addition, this approach is designed to take advantage of the desirable built-in features of the authoring language while maintaining the extensibility needed to implement new instructional designs.

Measuring CBT Developer Productivity

The major purpose of computer-based training (CBT) authoring systems is to increase CBT developer productivity. One standard measure of this productivity is the number of development hours needed to produce one hour of on-line instruction. Measures quoted in the literature typically fall between 50 and 150 hours of development for each hour of on-line instruction, but can range to as many as 300 hours for complex equipment simulations.

It is often difficult to reconcile these measures with claims made by authoring system vendors, who typically dramatically underestimate the time taken to produce highly creative, finished instructional products. When you hear claims that a certain authoring system allows developers to produce 50 screens per day, you can be assured that that includes neither adequate design or testing time nor the realities of the production environment, i.e., insufficient access to subject matter experts, equipment down time, etc. It is our experience that even the best CBT developers can produce only one or two instructional sequences per day, where a sequence is defined as a series of 6-10 related displays, each requiring a non-trivial response (i.e., requires answer analysis) and each evolving from the previous display without a full screen erase.

Productivity and Authoring Systems

Regardless of what you include when you tally up your development hours, the key to increasing developer productivity is to reduce the number of hours needed to produce each hour of on-line instruction. The most common means used in pursuit of this end is the adoption of a CBT authoring system. Such systems help enhance productivity in a number of ways. The list that follows highlights the most typical approaches. We have included comments on the benefits and shortcomings of each.

(1) A language-less system for specifying typical screen displays and their appropriate presentation sequence.

The main shortcoming of this approach is that it addresses only the most visible aspect of CBT: what students see on their screens. It severely limits the developer's ability to address the more significant aspect of on-line instruction: the implementation of meaningful student/computer interactions.

Some systems that employ this approach contain very nice graphics editors with special routines for screen wipes, animation, etc. In general, however, we have only found these systems to be useful in creating the most elementary of instructional sequences, e.g., those that demonstrate a process by displaying a linear sequence of "slides" and asking the student to press RETURN to go from one to the next.

Language-less systems are typically tightly "bounded," i.e., they do not allow programming outside of the features they supply themselves. As such, we have found these systems too restrictive for all but the simplest of interactions. Most of these systems are geared toward beginning developers, but some are now appearing with enhanced functionality for more sophisticated applications.

(2) A simplified language for specifying typical instructional sequences.

Simplified language systems were devised to reduce the programming skill needed to produce CBT courseware. Such languages typically lack one or more of the basic components of a computer language, e.g., the ability to declare variables, to create looping and if/then structures, or to segment a program via subroutines. Like language-less systems, these systems can be useful for beginners, but we have found that developers quickly grow out of them as they gain experience. Rather than trying to make developers into programmers, we recommend that organizations employ a team approach to courseware development, with experienced programmers assigned to support instructional and subject matter experts who are designing CBT materials (Heines, 1985; Heines & Moreau, 1980).

The benefit of simplified language systems is that most of them employ a very clean and easy-to-learn syntax for typical CBT constructs such as multiple choice questions. Their shortcoming, however, is that (like language-less systems) they are usually tightly bounded and severely restrict creative extensions. In other words, they make the simple trivial and the difficult impossible. We have found that it is usually easy to write our own templates or subroutines in sophisticated authoring systems to implement typical CBT constructs, thus gaining the main benefit of using a simplified language system without begin limited by its non-extensibility.

(3) A set of program templates for implementing typical instructional sequences.

A template is a program shell that is fully operational but lacks subject matter content. Templates are typically devised for authoring systems that do not have subroutine capabilities. They allow developers to implement their designs within the constraints of a predefined instructional sequence, simplifying the programming task.

The principal advantage of using templates over simplified languages is extensibility. If a developer requires a new type of instructional sequence, a programmer can usually implement a template to achieve the desired effect within a matter of hours. Course developers then need only insert their subject matter content into the the program code in the appropriate places to use the template in their courseware. Once a template is developed, it can be used any time an instructional design calls for its approach by plugging in the appropriate subject matter.

The principal disadvantages of templates are that they foster large programs by requiring code to be repeated each time it is needed and, consequently, make revision of such programs difficult because a change to a template requires changes in each section of code in which that template is used.

(4) A set of utility programs and subroutines for implementing typical features.

Alfred Bork and his colleagues at the University of California at Irvine write most of their CBT materials in Pascal. Bork has stated that they use Pascal "not because it's a good CBT authoring language, but simply because it's a good language" (Bork, 1981). If you look at Bork's code, however, you will see that it consists mostly of calls to subroutines that implement screen management, testing strategies, and many of his standard instructional sequences. Bork and his colleagues have built up a sizable library of utility programs and subroutines that they use as building blocks when creating their courseware.

This approach is, in fact, typical practice for virtually all experienced software engineers. One seldom writes a program completely from scratch, and most software specialists draw on a personal as well as public library of common routines to use as program building blocks. This approach provides both the simplicity and extensibility of program templates, but usually results in a smaller final program than the template approach. (The template approach requires a complete copy in each instructional sequence that uses it, while only one copy of a subroutine is needed regardless of the number of times it is used.)

The shortcoming of this approach is that most CBT authoring languages don't provide full subroutining capabilities with parameter passing and local variables. One therefore has to move to a general purpose language like Pascal and re-invent all of the features inherent in even simplified CBT authoring languages, such as response analysis, student record keeping, etc. More sophisticated features, such as author-defined variable pitch character sets, may even require the writing of assembly language routines, a tedious task even for experienced programmers. In addition, even relative simple code modifications can require the help of experienced programmers. Thus you might gain the benefit of good software engineering but lose the benefit of the efforts that have gone into developing the useful features inherent in CBT authoring systems.

The major trade-offs one must consider in choosing an authoring system are therefore simplicity vs. power and built-in features vs. extensibility. The next section describes a somewhat different approach with which we have been experimenting, the construction of an "instructional database." This approach attempts to minimize trade-offs and attempts to provide the best features of each of the four approaches described above.

An Instructional Database Approach

Scientific Systems, Inc., (SSI) is often asked to develop courseware employing standardized instructional sequences. One of the most common of these sequences is one in which students are asked to name the parts of a complex machine, locate them in a diagram or photograph, and state their functions. SSI has developed a standard approach to "part definition" task and has implemented its approach on both TICCIT and the IBM PC.

In the past, SSI course developers have researched the subject matter, specified the names, locations, and functions of the machine parts to be taught, and provided this information to programmers for course implementation. Since the developers write their specifications on-line, SSI wished to devise a method for automating the specification-to-program step and eliminate redundant data entry and programming. Our solution was to design an instructional database. Using this approach, developers enter their course specifications in a simple format using any standard word processor. Their raw specification files are then read by a program that converts them into a form which can be read directly by a running CBT program. The program that reads the raw specification files functions as a "front-end" to the CBT program, allowing the CBT program to access any part definition record in the instructional database with a surprisingly small subroutine.

The beauty of the instructional database approach is that it provides both simplicity and power. Course developers see the simplicity of the system by working with it at the database creation level, which can be accomplished with any standard text editor. They can use the front-end program to check their raw specification files for syntax errors and see how their displays will look when processed by the CBT program. If they are not satisfied with the results, they can make corrections and/or alterations in their original files by re-editing their files with the text editor of their choice.

Programmers see the power of the system by working with it at the database reading level, after it has been preprocessed by the front-end program. The information in the database can be used in a large number of ways. For example, the same information can be used to generate:

- linear slide show-like sequences for introductions and reviews,

- help sequences that explain the location and function of a named machine part,
- practice exercises that require students to indicate the location of a part given its name or function,
- practice exercises that require students to supply the name of a part given its location or function, and
- test items that match stated functions with part names and/or locations.

The instructional database approach also provides a wealth of built-in features with complete extensibility. The front-end program is written in Pascal, and we have implemented versions that run under both VMS on VAX systems and DOS on IBM PCs. The database reading program is written in TenCORE, a powerful PLATO-like authoring system for the IBM PC available from Computer Teaching Corporation in Urbana, Illinois. We can therefore take advantage of all of the authoring system's built-in features because the courseware is ultimately presented using by that system. In addition, we can add our own features (like automatic text centering and generation of lists of bullets) by modifying the front-end. We can also create new instructional sequences by writing new subroutines to use the information in the instructional database in novel ways.

Productivity and the Instructional Database

The instructional database approach has completely eliminated the need for specifications written by course developers to be retyped by programmers. Once the specifications are run through the front-end program, at least one aspect of the course is completely ready for presentation without further programming. Our programmers thus concentrate on providing a library of standard routines rather than implementing each instructional sequence designed by the developers. When developers do specify a new design, that routine becomes part of the library and can be used for any subject matter.

Thus we believe that we have achieved the benefits of both worlds by employing an approach that is easy for non-programmers to use but does not inhibit their creativity. At the same time, the approach is very efficient from a programming point of view and readily extensible to accommodate new designs. We are now working on building our library of standard routines and assessing the full impact of this approach on the number of hours needed to produce each hour of on-line instruction. If the results of these efforts are as positive as we anticipate, we hope to achieve very significant increases in CBT developer productivity for a wide range of instructional applications.

References Cited

Bork, Alfred, 1981. Learning With Computers. Digital Press.

Heines, Jesse M., 1985. Anybody Can't Do CBT. Training News 6(7):9.

Heines, Jesse M., and Ken Moreau, 1980. The Three-Pronged Computer-Based Course Development Process. ADCIS Proceedings, Washington, D.C.

JOB AIDS: GIVE THE READERS WHAT THEY WANT

by

Patricia Lawson

ADVANCE PROGRAM

Technical communicators increasingly find themselves writing "job aids". This presentation defines job aids, discusses their applications and effectiveness, presents hints on writing them, and warns about their shortcomings.

ABSTRACT

More and more, technical communicators, especially those in computer-related fields, write so that their readers can do. Yet manuals are unopened and of those that are, much of their content is left unread. What readers often need is bare-bones, procedural information in an accessible, compact form - a job aid.

Fundamentally, job aids are like cheat sheets. They tell what to do, how to do it, the order of execution and the standards expected. They are succinct, procedural, visual, text-based, tangible and portable.

Job aids are for use on-the-job, when more than seven steps are involved, for skills areas rather than knowledge or attitude, for use as a checklist, as a supplement to training, where reference material is not accessible, where accuracy is critical, with infrequent procedures and/or for particular applications.

Job aids are cost-effective. They can enforce on-the-job accountablity and reduce errors and their associated costs. Since they can be used instead of or as well as training, they can reduce instructor costs and off-the-job trainee time.

The key elements of writing good job aids are 'observe, talk and do'. Just like all technical writing, job aids should be tested with a sample target group and updated regularly.

Users concerned with the performance of one-the-job procedures like job aids and use them effectively for the right applications.

JOB AIDS: GIVE THE READERS WHAT THEY WANT

What is the first thing most readers do when they must read a manual to do a job? They scan the section that explains the task and write "cheat sheets." These are notes that contain only the bare-bones, step-by-step procedures for getting their particular job done. Then they put these notes in a handy place and refer to them without ever looking at the manual again. Job aids are basically formal "cheat sheets."

More and more, technical communicators, especially in computer-related fields, are asked to write legitimate, accurate "cheat sheets." This is because we often deal with readers who don't care about why or how some piece of equipment works. They are primarily concerned with what they have to do to get their job done. They are interested in performing, not learning to perform.

This presentation defines job aids and their applications, discusses their cost and performance effectiveness, presents some hints on creating them, and warns about their shortcomings.

Definition

Job aids have been around for a long time, under various names. The term job aid is a collective noun phrase to describe a genre of tools that aid performance. Their purpose is to specify what is to be done, how to do it, the order in which to do it, and/or the standards to be met.

Perhaps we can better define job aids by stating what they are not. They are not, in the truest sense, instructional. That is, users are not supposed to learn the procedures by heart as a result of reading the job aid. Users are supposed to refer to the job aid every time they perform the procedure. They use the job aid as a reminder and as a checklist, not as a "teacher".

Because job aids are not instructional, they do not take the form of reference material, tutorials, explanation or detail. They do take the form of diagrams, checklists, algorithms, flip charts, on-line helps, etc.

If job aids can take so many forms, what do they have in common? Fundamentally, job aids are succinct, procedural and usually visual. In addition, they can be text-based, tangible and portable.

Job aids are succinct; they use the barest number of words to convey only essential information. Nice-to-know, explanatory information belongs in a reference manual. Job aids refer to a specific application and limit their information to that application.

Job aids are procedural; they list step-by-step instructions; each contains one action. For example, "open the disk drive latch and insert the boot diskette" should be two steps.

Job aids are usually visual; they contain as many visual cues as possible. These can include color diagrams, cartoons, mechanical drawings, graphics, etc. The visuals are as important as, if not more important than, the words.

Job aids can be text-based. The text may or may not be in sentence form. It is rarely in paragraph form. The words and structure are clear - they use no complicated vocabulary and no complicated phrases.

Job aids can be tangible. They are appropriate to the job and physical environment in which they are to be used. They are meant to be pointed at, manipulated, gripped with greasy hands, shoved in dirty nooks and crannies or in other words, "manhandled" in a hostile environment. So they'd better be durable too.

Job aids can be portable. They are meant to be used wherever the task to be performed has to be done. That can be beside a computer terminal, up on the wall, on-line, inscribed on a metal plaque, printed on a warning label, or stuffed in a pocket or briefcase and carted about. Job aids should be easy to carry around (if necessary), easy to consult, easy to use: in a word, accessible.

The most successful example of a job aid is the instruction plate on self-serve gas pumps. Fifty million drivers became self-sufficient within five years without courses or manuals. Other everyday examples of job aids include cookbooks, laundromat instructions, telephone books and dictionaries.

Applications

Okay, now that job aids are defined, how about their application? They certainly aren't good for everything.

a)For Use on the Job

Job aids are designed for use right on the job. An example of this is the manual flight "computer" used by pilots for pre-trip and in-flight computations. Flight "computers" have instructions and formulas printed right on them. These handy tools can be used where they are needed - at home or in the plane.

b) For More than Seven Steps

Most people can remember five to seven steps. For a procedure that requires more than that, they should refer to a job aid to verify that what they are doing is correct. Mandatory job aids also ensure that safety procedures will be followed.

c)For Skills, not Knowledge or Attitude

Job aids are useful when the procedure requires skill, not knowledge or attitude. In other words, the performer is being shown how to do something and is not expected to remember the steps or change their attitude as a result. For example, at Microtel Learning Services we produce computer-based training on a number of delivery systems. In order to take the courses, the student must be able to use the computer system to sign on, sign off, access the information, input answers and move around in the course. This information is secondary to the information they are supposed to remember from the course. We have created a series of job aids to help them use their delivery computer system particular to their needs as a student.

d)For a Training Substitute

As with the gas pumps, job aids are used where formal training or lengthy documentation is not appropriate. What's needed is a reminder of what to do and in what order so that the doer can perform the procedure accurately with a minimum of error and frustration.

e)For a Training Supplement

Conversely, job aids can be an inexpensive way to ensure that training carries over into the workplace. This application occurs when procedures are lengthy (more than seven steps), critical or site-specific. Job aids used in the training process provide a structure (a starting place) on which to "pin" reference information. In this way, job aids can act as a check list.

f)For Inaccessible Reference Material

Big manuals and sets of information are cumbersome to use on an ad hoc basis. They're heavy and bulky. It can be awkward to find information in them quickly. Job aids are meant to be easily carried around and set up at the job site for ready reference.

g)For Accuracy

When order of execution and accuracy are critical, job aids provide an essential cross-check. For example, the U.S. Navy issues checklists on routine maintenance procedures. Even though those sailors may "know" the routine, they are required to systematically follow the checklist that has been developed for every maintenance procedure (no matter how simple).

h)For Infrequent Procedures

Job aids are useful when the procedure is performed occasionally (less than once a week), where the performer can't be expected to memorize the process. In this case, the job aid serves as a memory job, a reminder.

i)For Particular Applications

Certain applications are customized for a particular site. In these instances, the procedures may vary slightly from the generic. A job aid to detail the particular local procedures for each job site is helpful. For example, how many versions of DOS do you use? Each one of them applies the commands a little differently. I have made my own "job aid" to carry out the few commands I have to use in my current version of DOS.

Cost and Performance Effectiveness

Since job aids can be used instead of formal training for simple, straightforward procedures, they can eliminate the cost of an instructor while reducing the learning curve of trainees who must use new techniques. Trainees don't have to memorize critical information; they learn how to use the job aid effectively. This reduces training time.

Since job aids can be used to reinforce formal training, they can reduce both instructor and trainee time in the formal training situation. Both of these situations mean definite cost benefits in terms of reduced instructor hours and off-the-job time for employees.

In addition, because job aids also act as a checklist for critical procedures, they can enforce accountability and reduce on-the job errors and the associated costs.

Creating a Job Aid

Okay, now we know what job aids are and where and why to use them. I'm sold. But how do you make them? How can you tell if they are any good?

a) Plan the content.

"Job aids are designed after careful definition of the [performance] objectives, taking into consideration production facilities and cost."(2) Ask yourself, "What do I want my users to be able to do while using this job aid?" This determines content.

In terms of utility, ask yourself "Can the job aid become part of the worker's daily environment? Will the worker be likely to use the job aid when confronted with the associated task? Is the task so critical that there is a significant risk involved if the worker does not use some type of job aid?" (3)

Having satisfied yourself with the answers to these questions, get on with the writing. The key elements of writing good job aids in terms of content are 'observe, talk and do'.

b) Watch people perform the procedure.

And I do mean people (not a person or two) - as many as possible because they all will do it a little differently. Note the order, the speed, the effectiveness and the difficulties of each person. Think about the reasons for these variations. Are they significant enough to be covered in the job aid?

c) Talk to the people afterwards.

"Afterwards" is important; otherwise, your "actors" may feel self-conscious which will affect their performance. Make sure that you talk to a cross-section of people, from beginner to expert. The most reliable source is a good practitioner - the one who has been at the job for a year or two but who doesn't know the job so well that basic and prerequisite knowledge and skills are assumed.

d) Do the task yourself.

There is an old adage which says, 'never ask anyone to do anything that you haven't tried yourself.' You have to understand the task from the doer's point of view. Think about the problems you have in doing the task and write to address these. There are exceptions, of course, e.g. writing the landing checklist for a 747.

e) Test the job aids.

This ensures that they are accurate and useful. Also, "it is important to keep job aids up-to-date as jobs and methods change."(4)

Shortcomings

Despite all of these positive aspects of job aids, be aware that they are not a panacea to training and performance concerns.

Such things as color, die-cut tabs, plasticized cards, heavier stock paper, easel-type binders, etc. make or break some job aids. These customized features can cost a lot to produce, no matter the quantity. This problem accelerates when procedures change often. However, for the right applications, job aids very quickly pay back these costs in terms of reduced instructor and trainee time, as well as reduced error costs.

Job aids are only useful in skills areas where speed is not the essential element of maximum performance. They are not useful in areas where knowledge or attitude is critical to performance. A job aid would retard split-second decisions required in critical knowledge areas, like emergency landings.

Job aids are also not useful for complicated procedures with lots of options. Neither should they be used instead of training where long-term learning is the desired result.

Conclusion

You can lead readers to content but you can't make them drink. As writers of technical manuals, we usually write so that people can do. Yet most manuals are unopened. Of those that are, most of their content is left unread. If most readers do not read all that "valuable explanation," why not give them the information they want in an accessible, compact form? If you want to be their hero, create a job aid. Most people like job aids and perform effectively when using them for the right applications.

REFERENCES

1. Kershner, William K., The Student Pilot's Flight Manual, Third Edition, Iowa State University Press, 1968, p. 29.

2. International Telecommunications Union, Reference Manual for Telecommunications Training Centres, Provisional Edition, June 1980, p. 49.

3. Lowell, Yarusso, "Job Aids - Panacea or Placebo", Performance and Instruction Journal, National Society for Performance and Instruction, October 1984, p. 1.

4. op.cit.

Contact:

Patricia Lawson
Microtel Learning Services
1795 Willingdon Avenue
Burnaby, B.C. V5C 5J2
Canada

604-293-3653

ADCIS Proceedings, November 1986

DESIGN AND RESEARCH ON THE INTERACTIVE ENCYCLOPEDIA SYSTEM (TIES)

Janis Morariu
The University of Maryland University College

Ben Shneiderman
The University of Maryland

ABSTRACT

The Interactive Encyclopedia system (TIES) has been under development at the University of Maryland since 1983. Through the authoring system, an author can create a database of intricately networked articles and images on any topic. The browser system allows learners to explore the articles and images by touching the screen (or using arrow keys) to access brief definitions on a topic highlighted within an article, and retrieve the full article on the identified topic. Also discussed are the instructional design of TIES from a human factors perspective and five experimental studies of system features.

The Program

The Interactive Encyclopedia System consists of the following components: (1) authoring software for composing, linking, and editing databases of new articles on any topic, (2) browsing software for learner traversal of databases of interwoven articles, and (3) databases of linked articles for user browsing. TIES runs on IBM PCs, XTs, and ATs with monochrome or color monitors, and with or without touchscreens. TIES has been under development at the University of Maryland since Fall 1983 through the support of the Human/Computer Interaction Laboratory (Department of Computer Science), the College of Library and Information Services, and the Department of History of the University of Maryland, as well as the Center for Instructional Development and Evaluation (University of Maryland University College), Cognetics Corporation, and the United States Holocaust Museum Council (U.S. Department of the Interior).

This software program allows learners to explore information resources through "embedded menus" by merely selecting highlighted topics from within article entries. While reading an article on French Painters, for example, the learner may choose to read a brief definition of Impressionism, view some paintings by Degas, read a full article on Van Gogh, and/or browse through the index of related topics. As learners traverse articles, TIES records the path and allows easy reversal, giving learners a sense of confidence and control.

TIES allows authors to create a network of conceptual knowledge in which concepts are linked associatively, and the learner is free to explore pathways based on their needs and interests. It is an engaging challenge to reformulate pedagogic plans into the network of related articles and see the linkages come to life as they are used by students in novel ways. The TIES authoring software guides the author in writing a title, brief definition (5 to 25 words), text (50 to 1000 words, typically), and synonyms for each article title with possible links to computer-generated graphics and videodisc images. The author marks references to other topics and images in the text of the article with specified characters. TIES collects all marked references, prompts the author for synonym relationships, maintains a list of articles and images, and allows editing, addition, and deletion of article contents. The author tool displays TO/FROM citations for each article and allows authors to keep notes on each article. A simple word processor is embedded in the authoring software, but authors can create articles on their own word processor, if they wish. There are no commands to memorize; every operation is done by selection from options on the screen.

Potential applications include tutorials about software products, online manuals, museum exhibits, self-study guides for any discipline, travel guidebooks, annotated Shakespeare or the Bible, maintenance manuals for equipment, and adventure stories. Each user suggests intriguing and novel applications.

Empirical Studies

Five experimental studies have been conducted to test certain design alternatives (such as demonstrating the advantage of arrow keys over the mouse for this system) and observe user behavior. More than 220 subjects participated in these controlled experiments. In addition, more than three hundred novices and experts have tried and commented informally on the current design.

In a study comparing the arrow keys (maybe better termed "jump" keys because the cursor would jump to the closest target in the direction pressed) to the mouse, the arrow-jump keys proved to be an average of 15% faster and preferred by almost 90% of the subjects (Ewing, Mehrabanzad, Sheck, Ostroff & Shneiderman, 1986). When there are a small number of targets on the screen and when arrow-jump keys can be implemented, they provide a rapid, predictable, and appealing mechanism for selection.

In a second study using the TIES technique, subjects traversed a database with 42 articles about the University of Maryland Student Union (Powell, 1985). The embedded menus technique reduced the number of screens viewed when compared with an explicit menu strategy. There were significant reductions in the times for task performance, and the subjective preference was strongly for the embedded menus.

The embedded menus approach was also used for two experiments with online maintenance manuals for electronic equipment (Koved & Shneiderman, 1986; Koved, 1985). A tree structured and linear form of a 52 page maintenance manual was prepared for screen presentation and in paper form. Experimental subjects had to perform 12 tasks using one of the manuals. Significant differences were found showing that time was reduced using the paper versions. No significant differences were found between the tree and linear versions for speed or error rates. When a pruning algorithm was applied to the text to allow users to trim text unrelated to their task, the time was cut in half. This latter experiment used only the computer condition and demonstrated one of the advantages of screens over printed text. This is important, since for many applications printed manuals are still easier to use and approximately 30% faster to read than computer displays.

Four selection strategies (mouse, arrow-jump, touchscreen, and numbered keys) were compared in a study in which each of 24 subjects used each strategy to follow prescribed paths through the database (Ostroff, 1986). The touchscreen resulted in the fastest time and highest preference, but also the highest error rate. Improved strategies for touchscreen and the mouse were proposed.

Future Directions

In June 1986 TIES 2.1 was completed which contains support for display of graphic images. Current implementation efforts focus on support for videodisc and integration with other software. Further testing is needed to select the optimum touchscreen or other input devices, and to test alternate screen layouts.

An advanced browser with string search, various bookmarks, multiple windows, user annotation, printing, etc. is being designed. An advanced authoring tool with automatic reference marking and sophisticated editing is also planned. Present efforts also include implementing TIES on larger screen displays to allow multiple windows and pointing at graphics objects.

Availability

The University of Maryland has contracted with a software development firm to support commercial applications and the distribution of TIES. To find out more, contact:

Charles Kreitzberg, President
Cognetics Corporation
55 Princeton-Hightstown Road
Princeton Junction, NJ 08550
tel.: (609) 799-5005

References

Ewing, J., Mehrabanzad, S., Sheck, S., Ostroff, D., & Shneiderman, B. (1986). An Experimental comparison of a mouse and arrow-jump keys for an interactive encyclopedia. International Journal of Man-Machine Studies, 24(1), 29-45.

Koved, L., & Shneiderman, B. (1986). Embedded menus: Selecting items in context. Communications of the ACM, 29(4), 312-318.

Koved, L. (1985). Restructuring textual information for online retrieval. Unpublished master's thesis, University of Maryland, Department of Computer

Science.

Ostroff, D. (1986). Selection Systems: Interactive Devices and Strategies. Unpublished master's thesis, University of Maryland, Department of Computer Science.

Powell, D. (1985). Experimental evaluation of two menu designs for information retrieval. Unpublished manuscript, University of Maryland, Department of Computer Science, College Park.

Weldon, L. J., Mills, C. B., Koved, L., & Shneiderman, B. (1985). The structure of information in online and paper technical manuals. Proc. Human Factors Society - 29th Annual Conference (pp. 1110-1113). Santa Monica, CA.

TIES Authoring System - User's Manual, Version 2.0 (February 1986), 51 pages.

TIES Authoring System - User's Manual, Version 2.1 (Graphics), (June 1986).

Contact:

Janis Morariu
Center for Instructional
Development & Evaluation
University of Maryland University College
University Blvd. at Adelphi Road
College Park, MD 20742

301-985-7380

ADCIS Proceedings, November 1986

Tools for Managing CBT Courseware Development

Timothy W. Spannaus, Ph.D.

Tim Spannaus and Associates

ABSTRACT

Spreadsheets and project management software can significantly help the courseware development manager estimate time and budget for projects and control the projects once underway. Project management systems are designed to compared actual to projected outputs, and make changes accordingly. Spreadsheet templates are most appropriate for small projects; project management software comes into its own on large projects. Examples are provided of typical uses of both.

The purpose of this paper is to illustrate several computer-based techniques for managing computer based training development. I will describe some fundamental concepts of project management, then illustrate uses of spreadsheets and project management software for this application.

Project management basics

There are certain characteristics that set project management apart from other kinds of management. In general, these are limitations on the scope of the management task, by time and by objectives.

Limited by time

Projects have a specified duration.

They do not (or should not) continue forever, and need not provide for events beyond the life of the project.

Limited by objectives

Project management is concerned with producing one or more deliverables, not all of the activities that other managers have to deal with on a continuing basis.

Factors we control

Of these factors, quality is the most important. Time and budget are soon forgotten. But don't let schedule and budget overruns become a habit, or you won't have time to worry about quality.

Budget

Expenses, both variable and fixed, associated with a project.

Schedule

Time required to complete a project. You should consider both working time and calendar time.

Quality

Quality is more difficult to measure than budget or schedule. Quality control procedures, used consistently, are your best assurance of quality.

Tools for Managing CBT Development

Control systems

Management systems depend on feedback. The system measures the output of the process, compares it to what it ought to be, and makes changes if necessary to the inputs or the process. (See figure 1)

Sensor

Measures or observes the outputs.

Comparator

Compares real outputs to the desired outputs.

Actuator

Changes the inputs or processes depending on the difference signal from the comparator.

Tools

The spreadsheet templates and project management software described here essentially function as the comparator. The human project manager provides data (sensor) and makes necessary changes (actuator).

Spreadsheets

These all purpose tools help in estimating a job and in controlling small ones.

Estimating: Accurate estimates of time and budget are essential for proposals and feasibility studies. (See Figure 2)

Job costing: Job costing is the function of tracking actual expenditures for a project. (See Figure 3)

Project management software

This class of software is far more powerful than spreadsheets for project management, but are far more time-consuming to use. They are most useful for large projects.

Projecting schedules & budgets: You can use project management software for estimating, but you can probably do as well with a spreadsheet, especially for relatively small projects.

Tracking schedules and budgets: For complex projects (more than twenty or thirty tasks), use project management software. Projects have a way of getting out of control without accurate, timely reports.

Basic steps

Listed here are the basic steps that we perform in planning any project.

Break down the tasks

Analyze and list all the tasks to be performed. You can more accurately estimate time and budget for smaller tasks than larger ones, so continue breaking down until you have tasks of small enough size that you feel comfortable estimating time required to complete the task. Generally, this size task should be such that time estimates are on the order of ten days or less, rather than several weeks or months.

Estimate time and budget for each

Use a weighted formula to establish time requirements. Budget is generally a function of time (labor expenditures), plus materials costs.

The weighted formula for time is $[O + 4L + P]/6$, where O is the most optimistic estimate, L is most likely, and P is most pessimistic estimate.

Establish dependencies

Dependencies are those tasks which cannot start until another task is completed. For example, you cannot edit a manuscript until it is written, so editing is dependent upon writing.

Dependencies are depicted in a network diagram, or PERT chart.

Add in contingencies

Contingencies are those unforseen circumstances that affect time and budget estimates. The weighted formula allows for errors of estimating within a task. The contingencies to add here are additional tasks that you may have to include, such as would be required by changes in scope of a project. Do not include contingencies at full estimated time and budget, but weight their inclusion by the probability of each contingency occurring.

Add in overhead

Overhead expenses are those which do not directly produce product, but are an expense of doing business. Office space, clerical help, and the like, are overhead expenses which must be included in estimates that cover all expenses. Overhead is usually reckoned as a percentage of direct labor charge. Overhead may also be shown as a "burden" rate on labor.

Using the tools

In this section of the presentation, I will give examples of use of both spreadsheets and project management software, as they are used to manage budget and schedules for CBT projects.

Spreadsheet templates

Spreadsheet templates are simply spreadsheets with formulas already entered. The spreadsheets are used here to quote and to track budget on small projects. These templates are not as accurate as project management software, but are much faster to use.

Quoting: The quoting template works from estimated specifications for a CBT project to costs and sell price. Factors which will vary from one shop to another include labor costs, burden and markup, and productivity factors, such as weighting for different types of screens and numbers of screen produced per day.

Tracking budget on small projects: The budget tracking template assumes linear expenditures over the life of a project, and allocates portions of the total budget based on the ratio of weeks elapsed to total weeks. It compares actual expenditures to the linear estimate of budgeted expenditures and reports percent of budget for each cost category and for total budget.

Project management software

Project management software is an application especially design to use the network diagram, task breakdown, and time and cost estimates developed earlier to estimate total project duration and cost, and then to compare actuals to planned during the life of the project. Some packages will level use of resources automatically, while others will flag overcommitted resources. Those that level resources will reschedule tasks to avoid scheduling the same person to two simultaneous full-time tasks.

Application: These products, because of the detailed information you have to develop and input, are most effectively used for large projects. On small projects, you can develop and maintain a paper-and-pencil schedule and progress-to-plan reports. On large projects, recalculating schedules and critical paths can become so burdensome that many managers don't update the schedules to reflect actual dates. Use of project management software removes the repetitive calculations and highlights problem areas.

Weaknesses

All of these packages require detailed planning and estimating. None will be easy to use for those who have never used project management tools of any kind. Depending on the software used, it may be difficult to assign one person to several tasks, each of which is part time. It may be difficult or impossible to establish partial dependencies, in which a portion of a task must be complete before a subsequent task can begin.

Example

We will use this example project to illustrate the uses of the software.

The spreadsheet templates and project management reports are based on this project.

One hour CAI lesson

Specifications

- 60 screens
- 20 questions
  - 10 mc or short answer

- 10 graphic simulations
  (five of those are reused)
- 40 expository
  - 15 principally text
  - 30 text and graphics
    - 10 difficult
    - 10 moderate
    - (some of each are re-usable to make 30)

Tasks

- Start of work meetings
- Content research
- Flowchart
- Client approval
- First draft
- Edit
- Internal reviews
- Client reviews
- Second draft
- Edit
- Internal reviews
- Client review
- Pilot test
- Revise
- Client acceptance

Contact:

Tim Spannaus and Associates
22933 Coventry Woods Lane
Southfield, MI 48034

313-357-0004

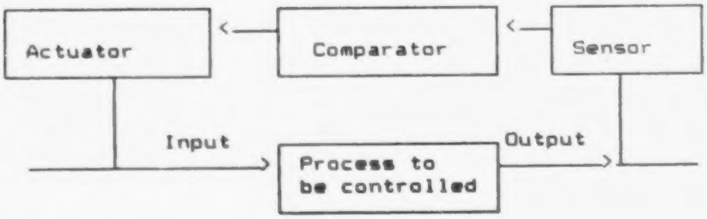


Figure 1: Management system

Quoting Aid

Specs	Quantity	Weight	Count
MC or Short answer	10	1.50	15
Graphics questions	10	2.30	23
Screen simulations	10	3.00	30
Text screens	15	1.00	15
Text and graphics	30	1.80	54
Totals	75		137

					Labor cost
Task	Quantity	Unit	Q/day	$/day	Cost
Start of work mtg	1	meeting	2	$465.00	$232.50
Content research	2	interviews	2	$135.00	$135.00
Flowchart	4	flowcharts	1	$135.00	$540.00
First draft	137	w screens	15	$135.00	$1233.00
Edit	137		65	$110.00	$231.85
Internal review	137		80	$180.00	$308.25
Second draft	137		45	$135.00	$411.00
Edit	137		120	$110.00	$125.58
Internal review	137		150	$180.00	$164.40
Test	137		10	$150.00	$2055.00
Revise	137		30	$135.00	$616.50

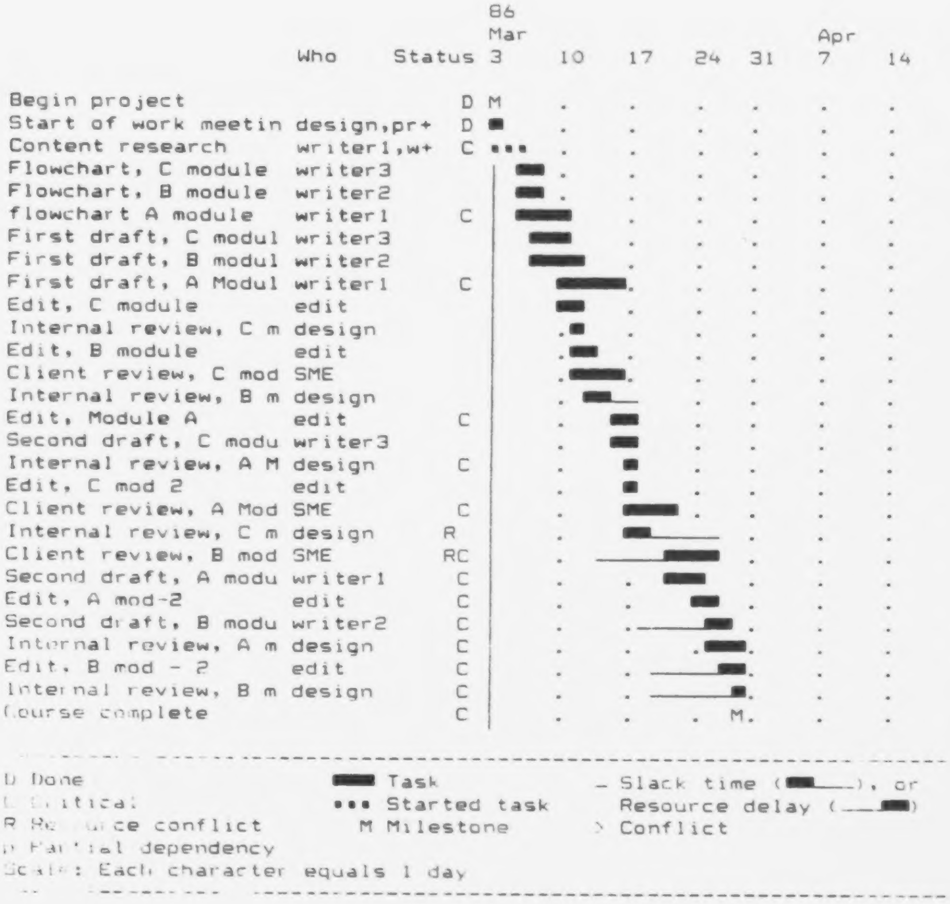
Sell price calculation	
Labor	$6053.08
Burden rate (%)	90
Burdened labor	$11500.85
Markup (%)	40
Sell price	$16101.19

Figure 2: Job estimating template

Report for week 3 of 4

Blitro Prodknow	Budget (total)	Budget to date	Total spent	act/proj (percent)	Week 1
Labor	$7125.00	$5343.75	$5629.00	105%	
(list names)					
Developers					
John M	$1300.00	$975.00	$1259.00	129%	500
Sue B	$1300.00	$975.00	$1085.00	111%	450
Raphael M	$1300.00	$975.00	$1125.00	115%	525
Design					
Gonzo	$650.00	$487.50	$465.00	95%	110
Edit					
B. Pencil	$750.00	$562.50	$550.00	98%	0
ProjMgr					
B. Counter	$325.00	$243.75	$270.00	111%	110
SME					
X. Pert	$1500.00	$1125.00	$875.00	78%	500

Figure 3: Job cost tracking template



	Who	Status
Begin project		D
Start of work meetin	design,pr+	D
Content research	writer1,w+	C
Flowchart, C module	writer3	
Flowchart, B module	writer2	
flowchart A module	writer1	C
First draft, C modul	writer3	
First draft, B modul	writer2	
First draft, A Modul	writer1	C
Edit, C module	edit	
Internal review, C m	design	
Edit, B module	edit	
Client review, C mod	SME	
Internal review, B m	design	
Edit, Module A	edit	C
Second draft, C modu	writer3	
Internal review, A M	design	C
Edit, C mod 2	edit	
Client review, A Mod	SME	C
Internal review, C m	design	R
Client review, B mod	SME	RC
Second draft, A modu	writer1	C
Edit, A mod-2	edit	C
Second draft, B modu	writer2	C
Internal review, A m	design	C
Edit, B mod - 2	edit	C
Internal review, B m	design	C
Course complete		C

D Done
C Critical
R Resource conflict
P Partial dependency
Scale: Each character equals 1 day

▬ Task
••• Started task
M Milestone

_ Slack time (▬__), or
Resource delay (__▬)
> Conflict

Figure 4: Gantt chart from project management software

BEST COPY AVAILABLE

ADCIS Proceedings, November 1986

RELATIONS: The Missing Link in Models of Cognitive Learning

Gerald J. Blumenfeld

The University of Akron

Abstract

Precise and consistent distinctions are needed between concepts and relations. Relations are classes of ordered sets and can subsume definitions of concept. Responses to relational stimuli involve transformations as hypothesized by Berlyne. Principles can be defined as rules for selecting and applying transformations. Since the theory requires dynamic stimuli and active responding, CBE can play a critical role in future applications.

The Missing Link

The word "relation" is used almost intuitively in most educational and psychological literature. When an author must use the word, relation is usually treated as a vague kind of concept. The authors of most articles implicitly assume that whatever variables affect concept learning operate in a similar manner when relations are to be learned. Therefore, it is not too surprising that, while one can find numerous entries under "concept" in the indexes of books on learning, it is difficult to find a book that even lists "relation" in the index. A similar fate befalls those who expect to find the same abundance of experimental work listed under "relation" in abstracts of psychological research that we find under "concept."

Yet, all of our reasoning ability depends upon the use of relations and their logical properties. The intellectual development of children depends upon the acquisition of both overt and covert responses to relational stimuli. This author assumes that good scientific research on the learning of relations will be facilitated by theoretical distinctions that lead to different operational definitions for concepts and relations. This paper will attempt to provide conceptual distinctions for both (a) the stimuli required when one attempts to operationalize concepts and relations, and (b) the responses to be observed and brought under experimental control when the learning of concepts and relations is studied.

An attempt will be made to synthesize some elementary ideas on relations from the field of logic with contemporary theorizing on the psychology of complex cognitive behavior. The resulting scheme will be used to encompass phenomena beginning with simple rote associations and then extended to suggest one way of distinguishing between concepts, relations, generalizations, and principles.

The relevance of these ideas to computer based instruction is two-fold. First, effective courseware depends somewhat upon the conceptions of learning and instruction available to the designer of the lessons. Secondly, the position advocated here is that dynamic stimuli as well as active learners are essential to the learning of relations. The modern electronic computer provides us with the delivery and control mechanisms necessary to achieve these essential qualities.

A Definition of Relations

Several years ago, the author read a study in which a generalization was defined as a relationship between two concepts. Concept was given the usual definition as a category or class of things. It was hypothesized that the learning of a generalization would be faster if both concepts included in the generalization were learned before the generalization was to be learned, rather than having to learn both the concepts and the generalization simultaneously. It appeared to be a reasonable hypothesis, and the reported data tended to provide confirmation. While thinking about this data, I wondered, "If having learned the concepts first helped, what about the relation involved?" Well, no consideration of this question was presented in the article. It was implicitly assumed that the relation was a given. While some current authors do not make this error (Engelmann & Carnine, 1982; Gagne, 1985), current treatment of relations involves much of the ambiguity and inconsistency of earlier years. Questions such as "What is a relation?", "How is a relation different from a concept?", "How can one operationally produce positive instances of a relation as one does with a concept?", are not readily answered. Eventually, these questions lead to the definition of relations used by logicians.

Relations such as "greater than" "south of" "under" and "sister of" are frequently used examples. A positive example of these relations requires the presentation of an ordered pair of things. Since two things are required, they are called dyadic relations. A dyadic relation then is a set of ordered pairs. Now, a concept is also a set of things, but each element of the set is a single entity. At least, that is how concepts have been defined logically and operationally in most experimental literature.

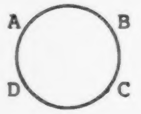
Dyadic relations can be represented symbolically as $\mathbf{R(X,Y)}$ where "R" represents a relational variable, and "X" and "Y" represent term variables. If we substitute Akron for X, Cleveland for Y, and "south of" for R, the proposition becomes "Akron is south of Cleveland." Infinite possibilities obviously exist. All dyadic relations can be represented by one relational variable and two-term variables.

Relations are not limited to sets of ordered pairs. While dyadic relations require two terms, it is possible for a relation to require three terms or four terms. "Betweenness" and "odd" are relations requiring three terms. These are called triadic relations. The statement "Mary gave that test to Joe" is also an example of a triadic relation. A triadic relation would be represented symbolically as $\mathbf{R(X,Y,Z)}$.

Relations involving four terms are not as common but can be represented also by $\mathbf{R(X,Y,Z,W)}$. These are tetradic relations. Most of us find it difficult to keep four things in mind at once. In fact, our ordinary language inhibits the use of tetradic relations. Lee (1961) gives an example of tetradic relations in her book on logic.

There are many language structures to express dyadic relations. The characteristic language structure in English for triadic relations is the use of the indirect object. There is no special language structure for tetradic relations, and when they are expressed, a circumlocution is necessary. Projective geometry uses a tetradic relation, for example, when, of any four points on a line, two separate and are separated by the other two. [On the circle in Fig. 1] the points A and C separate and are separated by the points B and D. This is a single relation between four points, and its nature is not altered by the condition that it is expressed grammatically as if it were two relations co-existing. Another example of a tetradic relation may be found in law, where, strictly considered, the contractual relation is tetradic: The party of the first part contracts with the party of the second part to do X for a stipulation Y. Unless all four terms are included the contract is incomplete (Lee, 1961, p.23).

Fig. 1



It is theoretically possible to have relations requiring five or more terms. Since the number of terms required to give a relation full significance is variable, Lee (1961) identifies this attribute of relations as their "adicity." Adicity is the minimum number of terms required to give the relation full significance.

The limiting case for relations would require only one term. Relations requiring only one term are monadic relations. "Bill swims" and "Columbus citys" are examples. These propositions can be interpreted as a special kind of dyadic relation. They can be translated as "Bill is a member of the class of those who swim" and "Columbus is a member of the class of cities." In this interpretation the form of the proposition is dyadic, and the relation "member of" holds between the two terms. Operationally, this is the kind of response we require when concept learning has been studied. We require the learner to decide whether one entity does or does not belong in a class. It is interesting to note that, while it has been common in educational and psychological literature to subsume relations under concepts, the monadic relation permits us to subsume concepts under relations. Concepts then become the first class in the theoretically infinite succession of relations, each with a higher adicity.

The stimuli of various phenomena studied in the psychology of learning can be represented if we vary the adicity of the relation and the number of unique objects that the terms can represent. A monadic relation would be related to simple discrimination if the one term of the relation can be represented by only one unique object. A concept is produced if we let the term refer to many objects of the same class. Paired associates are represented if each term in a dyadic relation refers to only one unique object. A dyadic relation is represented if each term refers to many objects. If all the objects of each term belong to the same class, the relation can be stated in the form of a generalization. Further consideration of this scheme will be given later.

Transformational Chains

It is time now to give some consideration to the nature of a relational response. Is a relational response to an ordered pair of stimuli of a different nature than is the response to either of the stimuli presented independently? A fruitful approach to this problem appears to come from the work of Berlyne (1965). Berlyne draws upon the theoretical tradition of Hull (1934), Osgood (1953), and Mowrer (1960) to translate Piagetian ideas into a Neo-associationist theory of thinking. These, and other theorists, have assumed that some fractional part of every sensory input can be conditioned to occur without the original situation reoccurring. This would apply to both externally and internally produced stimulation.

In Hull's (1934) theory the organism can come to represent covertly a series of originally external stimulus situations. Berlyne (1965) develops these covert representations and calls them "situational thoughts." He then argues that we can introduce between each pair of situational thoughts a "transformational thought." The assumption is that, in getting from the first stimulus to the second, S1 and S2, the organism had to do something. Something had to be done to transform S1 into S2. Now it is assumed that a fractional portion of this overt transformational response can be conditioned to occur implicitly and becomes a "transformational thought."

Inhelder and Piaget's (1964) evidence suggests that transformational thoughts develop later than situational thoughts. A simple example is the child's ability to draw the initial and terminal position of a stick that was held in a vertical position and then permitted to fall to a horizontal position. The child can represent these initial and terminal positions at an early age. Only much later can he represent the positions of the stick while it is falling.

Berlyne (1965) believes that transformational thoughts are necessary when perceiving or thinking about relations. When one responds to the actual ordered pair, some kind of transformational response occurs. This may be bringing two sticks together and feeling the hand or fingers move across their ends; or it may be only an orienting response as when one moves the eyes and head to focus on successive objects. It may be putting a "ball under a box." These overt transformational responses are considered to be necessary antecedents of relational thoughts.

Both Berlyne (1965) and Inhelder and Piaget (1964) believe that a child's ability to think about a class of objects, or to classify an object, is also dependent upon earlier motor activities in which the child has physically organized a set of objects into a group. Sorting objects into a group or groups is a transformation. It is assumed that classifying objects at the operational stage depends upon implicit transformational thoughts that originated by physically placing a new object into an established set. It is interesting to note that under the discussion of relations we considered concept as a monadic relation. Here we find that Berlyne also extends his assumptions about transformational responses from relations to classification.

Transformational thoughts are presumed to occur in at least two different kinds of habits. A transformation applying habit is required when the organism is confronted with a situational thought followed by a particular transformational thought. "Given this stimulus situation and this transformation, what stimulus situation will result." Transformation selecting habits are required when S1 and S2 are represented and the question becomes "What transformation will get me from S1 to S2?" It can be seen that the eduction of correlates and the eduction of relations problems used by Spearman (1927) in tests of mental abilities correspond to transformational applying habits and transformational selecting habits, respectively.

Implications

The scheme developed for concepts and relations can be extended to include principles and generalizations. Principles appear to be rules for selecting or applying appropriate transformations. One has only to consider many of the classic studies on thinking and problem solving to recognize the transformations that resulted in successful solutions. These include Koehler's studies with the apes (1952), Wertheimer's work on "productive thinking" (1959), Maier's experiments on reasoning (1930), or Katona's match stick problems (1940).

An example of a verbal expression of a principle is "the sum of N odd consecutive cardinal numbers beginning with 1 is N squared." This can be changed easily to "If you have a series like 1, 3, 5, 7, 9, and you want the sum, then count the number of cardinal numbers listed and square that number." The statement that, "A response that is not reinforced will extinguish" can be changed to "If you don't want a response to occur, then stop reinforcing it." Of course it is possible for one to apply the rule in a principle without being able to verbalize it. Generalizations were defined earlier as the relationship between two or more concepts. Examples would include "Boys grow into men," "Girls mature faster than boys," "Women live longer than men," "Elephants are larger than horses," and "Motorcycles are faster than bicycles." Generalizations do not appear to be rules for selecting or applying transformations. However, they might be transformed into principles or combined with other generalizations to produce principles.

As mentioned earlier, a variety of psychological phenomena can be represented by varying the adicity and number of objects that can represent the term. If we hold the specific instances of two terms constant and ask someone to make and explain successive changes in the value of the relational variable, we arrive at an example of creativity.

Empirical work in the area of learning relations should provide a great increase in our knowledge if at least two previously neglected factors are included in future concept learning type studies. First, relational stimuli should be employed, i.e., ordered sets. Second, any hypothesized transformational response needs to be made overt enough to come under experimental control. The static type of stimuli and relatively passive subjects of many concept attainment studies will be inappropriate. New paradigms will be needed to accommodate active, moving stimuli, and active, moving learners. The variety of input devices, graphics capabilities, and recording facilities available in CBE makes these requirements attainable.

Major problems in operationalizing these ideas have been avoided in this brief presentation. The following points appear to be critical problems. Berlyne posits natural relations and mediated relations. Natural relations involve intensities of attributes such as time, space, and weight that result in stimulus generalization. All other relational responses are mediated by association with natural relations. It may be quite difficult to find natural relations that are new to one's experimental subjects unless infants are observed.

Relations such as "greater than" apply to many kinds of stimuli. It may not be prudent to generalize results obtained with variations in light to variations in area. Tactile, visual, and auditory modalities present additional problems since the same verbal labels are assigned to relations in all three.

A final point involves the properties of relations. Dyadic relations have logical properties. Variations on these properties are likely to produce sets of ordered pairs that will vary greatly in complexity. For example, the equivalence relation is symmetrical, reflexive, and transitive; an ordering relation is asymmetrical, irreflexive, and transitive. Given the same matrix of stimuli from which to draw upon, very different sets of ordered pairs represent these two relations. This variation may be related to ease of learning.

An understanding of the learning of relations is essential to developing the most effective courseware. CBE can be an extremely useful tool in this quest.

Bibliography

Berlyne, D. E. (1965). Structure and direction in thinking. New York: John Wiley & Sons, Inc.

Engelmannn, S., & Carnine, D. (1982). Theory of instruction: Principles and Applications. New York: Irvington Pub., Inc.

Gagne, R. M. (1985). The conditions of learning. New York: Holt, Rinehart & Winston.

Hull, C. L. (1934). The concept of the habit-family hierarchy and maze learning. Psychological Review, 41, 33-54.

Inhelder, B., & Piaget, J. (1964). The early growth of logic in the child--Classification and seriation (E. A. Lunzer & D. Papert, Trans.). NY: Harper & Row.

Katona, G. (1940). Organizing and memorizing. New York: Columbia Univ. Press.

Kohler, W. (1925). The mentality of apes. New York: Harcourt, Brace.

Lee, H. N. (1961). Symbolic logic. New York: Random House.

Maier, N. R. F. (1930). Reasoning in humans: I. On direction. J. Comparative Psychology, 10, 115-143.

Mowrer, O. H. (1960). Learning theory and the symbolic processes. New York: John Wiley.

Osgood, C. E. (1958). Method and theory in experimental psychology. New York: Oxford Univ. Press.

Reigeluth, C. M. (Ed.) (1983). Instructional - design theories and models: An Overview of their current status. Hillsdale, NJ: Lawrence Erlbaum Assoc. Pub.

Spearman, C. (1927). The abilities of man, their nature and measurement. New York: The Macmillan Company.

Wertheimer, M. (1959). Productive thinking. New York: Harper & Brothers.

IMPLEMENTING AN INSTRUCTIONAL MANAGEMENT SYSTEM IN THREE K-12 SCHOOLS: A PROJECT REPORT

Dr. Michael L. Burger, Director
Learning Technology Center, Texas A&M University

Abstract

This paper discusses a joint project between the Learning Technology Center at Texas A&M University and three K-12 public school systems in Texas to develop a comprehensive instructional management system. The project began early in 1985 with needs assessment activities. Pilot testing of the first component of the system is expected to be completed by November, 1986. The design of the system is based on needs expressed by teachers in the project schools, and involves teachers and administrators in the design, development and testing of the system. An overview of project goals and activities, along with a project status report is included.

Introduction

The Learning Technology Center at Texas A&M University is working with three K-12 school districts to develop a microcomputer-based instructional management system (IMS). A major component of the IMS is supported by an electronic database of instructional elements which teachers can access to generate and evaluate lesson plans. In addition to assisting teachers in developing good, comprehensive lesson plans, the IMS automatically keeps track of which instructional elements have been included in lesson plans and therefore, can generate the reports that are required to meet recently mandated local and state accountability mandates.

The instructional design component is one of five components which are planned for the system. The other components will build on information generated from the instructional design component to help teachers determine their instructional effectiveness, student achievement patterns, and to help them plan and evaluate individualized instruction.

Pilot testing, which involved data entry (building the instructional element databases), began during the Summer 1986, and formal testing and evaluation of a prototype of the first component of the IMS is scheduled to begin during the Fall, 1986.

Discussion

Education is an information intensive enterprise, yet educators lack efficient methods of gathering, processing, and reporting data in sufficient detail to enable them to make effective instructional and management decisions and determine the impact of those decisions on students. But these capabilities are crucial before educators can be able to accept responsibility for and report the effectiveness of their instruction. These capabilities must be developed and implemented before the public can reasonably expect increased accountability from their schools.

Texas, along with several other states, has responded to pressures to increase educational accountability and has passed legislation (House Bill 72) to require teachers to demonstrate the extent to which their instruction matches state curriculum guidelines. However, no provisions have accompanied the legislation to help individual school systems or teachers become more accountable. Estimates indicate that Texas teachers (in approximately 1100 districts) will generate well over two million pages of information in attempting to comply.

It is our contention that even if all other aspects of education were "reformed", improvements, if they occur, will be imperceptible if no additional assistance is provided to educators to help them manage the vast amounts of information required to meet individual student needs and to demonstrate accountability.

Project Status

In January, 1985, the staff of the LTC began needs assessment activities in the three project schools to determine which instructional tasks and duties teachers wanted help with. Teachers were asked to identify (and rank order) the specific tasks which, if they had assistance with and could accomplish better, would result in making their jobs more satisfying and professionally rewarding. Besides being concerned with extra-class duties (e.g., lunch room supervision and bus duties), they identified tasks which are associated with managing instructional information (e.g., developing curricula and lesson plans, grading and tracking student progress, and completing reports) as those with which they desired the most help.

Using these data, the LTC has worked steadily with teachers and administrators in these schools to design and develop an IMS which addresses (in priority order) their concerns. Our plans are to develop the IMS in stages over a five year period, implementing it component by component, and building inservice and district planning activities around the development activities and vice versa. Teachers and administrators in each of the project schools are actively involved in determining the functions and operation of the IMS, including the design of the databases and user interfaces which will form a major component of the IMS.

Procedures and programs have been created to assist teachers in compiling curriculum information in forms amenable to the development of detailed lesson plans. These procedures are being used by teachers and clerical staff to not only assist in curriculum development efforts, but to also provide a framework for the building of the databases which form the basis for the creation and evaluation of lesson plans. In some instances, teachers are entering their own data, in others, they have placed their curricular information on paper documents which are then used by clerical staff to enter the information into the IMS. During the Summer of 1986, data are being entered in the following areas; fourth grade mathematics, junior high science and social studies, and high school biology and chemistry. Initial, single user pilot tests of the data entry procedures have been occurring during the Spring and Summer, 1986.

Lesson plans that can be generated through the use of a computer-based instructional management system (IMS) should prove to be more functional and more complete than those created using manual systems, because with the IMS, information which is important to good lesson planning is brought to instructors - they don't have to seek it out. In addition, time savings in making lesson plans, the increased ability to evaluate lesson plans, and the reduction in paperwork associated with creating lesson plans and in the production of required accountability reports can provide the necessary incentives to make automated systems work well.

Because instructional information is keyed into an electronic database as lesson plans and curriculum guides are being developed, teachers are simultaneously developing curricula, lesson plans, and an electronic database, which they can use for planning and evaluating instruction. Teachers who are using the system to enter their instructional information already recognize how the IMS can be used as a more effective and efficient tool for planning and evaluating their instruction than the paper-based documents with which they have traditionally worked. They perceive the time and effort needed to develop curricula and build their electronic databases as being offset by the advantages of, for the first time, having the information they need for good planning and reporting at their fingertips.

The computerized relational database which supports the system contains information on; (a) teachers, (b) students, (c) schedules, (c) subjects, (d) courses, (e) units, (f) objectives, (g) essential elements, (h) resources, (i) activities, and (j) strategies. A user interface displays relevant information to teachers as they plan lessons. From these displays, teachers can select those elements which they feel will produce the best lessons for their students from appropriate sets of instructional objectives, activities, resources, and strategies.

Timeline and Evaluation

Each of the project schools is developing a database of instructional information in mathematics and science which will be sufficiently complete for them to use in developing lesson plans and generating the required accountability reports by September, 1987. In addition to these features, we expect to be able to include some aspects of student grade recording and reporting in the pilot test. Initially, the grading and reporting function will be designed to "replace" grade books and to automate the calculation and reporting of student progress.

The activity of the teachers who are pilot testing the system has been effective in attracting the interest of other teachers who are already expressing a desire to "put their information into the system." We are expecting this interest to provide impetus for inservice activities and the corresponding "automation" of information in other subject areas. Initial development of the software and the building of the databases has been performed on IBM AT computers using PCDOS. Parallel pilot testing will occur on the IBM's and on Burroughs' Educational Microcomputer System - a QNX-based networked system which uses intelligent (80186-based) workstations linked to a fileserver. We calculate the data storage requirements to be in the range of one hundred megabytes per one thousand students.

During the first grading period, Fall, 1986, prototype systems will be evaluated by monitoring the use of the IMS and by obtaining biweekly reports from teachers who are using the system. Information will be requested that will enable us to determine whether or not the IMS is flexible enough, understood, easy and enjoyable to use, and whether or not teachers feel they are saving time and work by using the system. Routines imbedded unobtrusively in the IMS will allow monitoring of how much time is required to use the automated system to generate and evaluate lesson plans, as well as to obtain reports on which essential elements have been taught and which resources have been used. Data ascertaining time spent and teacher attitudes and concerns will be compared to similar data gathered from teachers who are not using the system.

Biweekly, administrators and teachers will be asked to rate the quality of lesson plans generated through the use of the system and to compare the quality obtained with the automated system with that obtained through traditional, manual methods of generating lesson plans. Assessment of quality will include; completeness, appropriateness, the extent to which the system is being used to generate lesson plans which serve as actual planning documents for both regular and substitute teachers, and an assessment of the perceived effectiveness in terms of student performance outcomes. Results of the evaluations and a more complete, up-to-date report on the project can be obtained by writing the author.

SEVEN LOCAL AREA NETWORKS

Jane A. Malloy

Delaware County Community College

ABSTRACT

Seven Local Area Networks have been installed utilizing the IBM PC Network Program for student instruction in engineering, business and graphic arts. The manner in which the network realized the administrative objectives of cost savings, security and user friendliness is discussed. In addition, the development and implementation of interconnecting hardware and software between the PCs and server is explained.

The IBM PC Network Program principally serves as classroom instruction about software programs in the fields of business, engineering, and graphic arts. It was installed at Delaware County Community College, Media, PA, last year. However, it was not used by the students for a semester so that the system could be "debugged". Instead, numerous demonstrations were given to the faculty and staff as they utilized the network.

Currently, there are seven networks in operation. Six are located at the main campus and one is at our nearby branch facility. Identical networks are used in the classrooms. Each comprise twelve IBM PCs and one IBM AT, server, which contains the software programs. Although the network instructions state that this system is capable of supporting seventy-two PCs, it has been found that it will not support this many in a classroom situation where the students are accessing the software and printer at approximately the same time. Our consultants and technicians recommend that this system be limited to twelve.

The faculty resource room and the president's office also contain a network with a different configuration than the classes. These utilize the IBM XT as the server which distributes information to three PCs.

OBJECTIVES

Before the College could purchase a network, much planning occurred in our Academic Computer Services Division which coordinates all the computer-related activities which include technical and media services, laboratories, and instructional resources. Since computers are located in various areas around the institution and used by all Divisions, careful study was made to determine exactly what the needs of the users were. After careful analysis of the classroom requirements of the Divisions, it was decided that networks should be placed in several areas. The main objectives to be realized by the networks were:

1. Cost savings
    - 1.1 Disks
    - 1.2 Printers
    - 1.3 Site licensing
    - 1.4 Communication links
2. Security
3. User friendliness
4. Electronic mail

A major reason for the installation of networks was the belief that there would be a substantial cost saving, particularly over the long run, by using networks rather than PCs alone. Prior to the installation of networks, disks needed to be given to each student to use for each program. Sometimes several would need to be handed out during the period. Our technician stated that it was like handing each student $1000. This distribution and collection of disks also took the time of the instructor and the lab instructional assistants who needed to keep a careful count of the disks to be certain they were returned after they were used. This resulted in each disk being counted at least four times per use. One can readily see that this could be very bothersome and costly when multiplied by the number of users and the number of disks.

After use, the disks were locked in a cabinet which was only accessed by the assistants to the supervisor of the laboratories. Despite all these precautions, disks were damaged and occasionally lost.

In addition, only one Hewlett Packard 2686 laser printer was needed to service the twelve PCs in the classrooms. The laser was chosen for its speed. However, letter-quality or dot matrix printers can also be used with the system.

Site licensing is another feature made possible by the network. At the moment, we are not taking advantage of this because, at least initially, enough disks had been purchased for PC classroom use. However, as the current software programs become obsolete and it is necessary to purchase new ones, the cost saving which is available from site licensing rather than purchase of individual disks will be substantial. In one classroom alone, twelve disks would need to be puchased if there were no network.

Currently, the network is not connected to the mainframe. However, long-range plans are to set up the network to share communication links with the IBM 4361 by using gateway program products. It is also hoped that word processing laboratories will be available for all students.

A major security problem was the possibility that the disks could be copied by the students. This has been eliminated with the network because a restricted-use password has been placed on the server so that the programs cannot be duplicated. Of course, nothing can be written on the network either. This is accomplished by eliminating the write command on the AT.

The network program is very user friendly. The screen shows a list of numbered menu options which the student selects by choosing one after inputting an identification number. When a software program is chosen from the menu, it is relayed to the PC and the student may begin using it.

With the net, the PCs can operate alone as well as by sharing the printers, disks and directories.

Another advantage of the PC Network is the electronic mail feature. This means that the PCs within a classroom can communicate with one another. They are able to do this by accessing each other on the screen by a number which has been assigned to each PC. The AT does not currently access the PCs so the instructor is not able to determine how the students are progressing with their work. However, the electronic mail feature becomes an excellent teaching device; the students learn how efficient it would be to be able to communicate in this manner. Of paramount interest to them is the way in which messages can be left for the person to access the system when he has time. They note how this could eliminate or alleviate interruptions when someone is in conference. It also tells the sender when messages are read and received.

DEVELOPMENT

Careful consideration about developing a microcomputer network began a little over two years ago when the number of software programs needed by the college began to multiply as computer laboratories spread. At the same time, the number of mainframes increased from one to three--two Hewlett Packard 3000s and one IBM 4361. These, however, are used solely by students in data processing courses and for administrative needs. They are serviced and programmed by on-site consultants.

The PC computer laboratories were set up largely because of community requests for courses in microcomputer use, particularly in the fields of engineering and business. Workshops were held at the campus and the computers were also taken to the sites where customized training was given. Concurrently, curricula and courses rapidly developed in remedial English and math, engineering technology, graphic arts, accounting, and in microcomputer use per se.

All of these many changes and increasing demands on the Academic Computer Services Division meant that networks could be readily justified because of cost savings, security, and user friendliness. The AT was chosen because its twenty-megabyte diskdrive could hold the needed software programs. The cost per classroom for the network was approximately $19,000 for the AT, laser printer network hardware and software (PC Network Program, DOS and blank diskette). The hardware included the adapter, translator unit, base expander, net short distance kit and coaxial cable.

IMPLEMENTATION

The network in each instance was connected by coaxial cable going around the room so that each PC was connected to the AT. This was accomplished by a broadband base expander which had nine terminals for attaching cable to eight PCs and one to add it all to the AT via the translator. The translator is attached to a wall and makes the systems of the AT and PCs compatible by interpreting the individual disk operating systems (DOS) to one another. Each classroom currently has three broadband base expanders in case the number of PCs should be increased. The attached diagram details how the hardware is placed.

A Local Area Network Adapter was attached to each computer. This adapter has a network cable attached to it which enables it to access and receive information. The network server is reached by a key diskette that contains the PC Network Program with the identification and other information that is required for interaction within the system. This key diskette is PC independent and network dependent. That is to say, it could be used with any PC in a network, but it must be recognized as an identified user which is programmed within the AT. Furthermore, each PC continues to act independently of the network.

This new technology created some problems. On several occasions one or more PCs locked up and the instructor was unable to continue. In other cases, some PCs were unable to establish contact with the network for no apparent reason. Additionally, diskettes would not work within the classroom although later testing indicated nothing was apparently wrong. At other times, files have been erased or diskettes written upon without apparent reason or explanation. Finally, the printer has stopped working or never worked during a class period.

This varied collection of difficulties has been resolved in some instances. One problem has been removed by not having all the PCs access the network or the printer at exactly the same time. Simultaneous use apparently created a multiplier effect which often meant the networks and printers would not operate. A form is now given to each instructor to complete after using the network; this is a check-off sheet which lists possible problems which could occur with space to add unusual ones or any comments. All the instructors' forms will need to be carefully monitored to determine if there is a pattern to the problems which have occurred.

It is believed that all "bugs" will be rapidly eliminated as more experience is gained in using the network. Then, the networks will be further expanded to include modems, communication to the mainframe, and greater access between the students and faculty.

CONCLUSION

The network overall has been a very positive addition to classroom instruction. However, continued use will be necessary to have it 100% problem free. Nonetheless, the objectives have been met. The cost saving has been substantial in part by not distributing the disks to the students. The risk of loss or damage to them has been eliminated as well as time saved by the instructors and assistants in not needing to count and care for the disks. The laser printer has alone been able to do the work of several dot matrix types. It is anticipated that site licensing and expanded communication links to the mainframe will result in additional cost savings.

The security of the software programs has been assured by the lock device placed on the AT so that none may be copied. Of course, the network menu program is easy to understand and use; the instructor need spend only a short time explaining how it is used. Furthermore, the electronic mail command simulates office use at present and will be used in the future for interaction between the students and faculty.

REFERENCES

Calkins, Mark (1985). Serving Up Servers. Micro Communications. June, 27-31.

Communications Family (1985). IBM PC Network Program User's Guide. Boca Raton: IBM, Inc.

Kolodziej, Stan Coping with Network Security (1986). Computerworld Focus. January 15, 37-40.

Sachs, Jonathan (1984). Local Area Networks. PC World. July, 69-75.

Thurber, Ken (1985). Departmental Solutions. Micro Communications. June, 20-25.

White, Lee (1986). Putting Network Management Into Good Hands. Computerworld Focus. January 15, 41-43.

Contact:

Jane A. Malloy
Jone-Malloy Associates, Inc.
Central & Lancaster Avenues
Berwyn, PA 19312

215-644-4200

DRILLS VS. GAMES--ANY DIFFERENCES
A PILOT STUDY

David W. McMullen
University of Illinois
Department of Secondary Education
1310 South Sixth Street
Champaign, Illinois 61820

ABSTRACT

This study investigated the effect of informational, drill, and game format computer-assisted instruction (CAI) on the achievement, retention and attitude toward instruction of sixth grade science students. An informational CAI lesson on Halley's Comet was administered to three randomly selected groups of sixth grade students. A CAI drill about the content of the informational lesson was given to one group, a CAI game was given to another group, and one group received only the informational lesson. No significant differences were found between the groups on a posttest measuring achievement given immediately after the instruction or on a retention posttest given one month later. Students in the group receiving the CAI game did differ significantly from the other groups in their attitude about how much they learned from the lesson.

Microcomputers are becoming commonplace in science classrooms--so the question is not whether to use them, but how to use them to deliver instruction effectively. Computer programs exist which provide information, drill and practice, tutorial instruction, simulation of real-life situations and educational games. Each of these methods, while different in style and format, is capable of presenting the same content to students. Does the method of delivery have any effect on learning or attitudes of students toward instruction? The purpose of this study is to investigate the effects of informational, drill and practice and educational game format of CAI on achievement, retention and attitude.

In the 1970s, computers were seen as having vast potential for the teaching of science--especially in the areas of simulation, interactive instruction, and individualized study (Lunetta & Dyrli, 1970). Lipson (1983-1984) recommends the use of intelligent drill and practice programs in the science curriculum. Ellis (1984) advocates using educational games to motivate students to master basic science facts.

Numerous studies have compared the effectiveness of CAI with traditional instruction--instruction not involving the use of computers. Edwards, Norton, Taylor, Weiss, and Van Dusseldorp (1975) reported that CAI instruction resulted in equal or higher achievement in a majority of studies. Zsiray (1983-1984) reported that CAI was at least as effective as the lecture method and significantly more effective than independent reading. Kulik, Bangert, and Williams (1983) concluded that students learning with CAI performed better than those learning by traditional methods.

The effectiveness of CAI drill and practice versus traditional instruction has been demonstrated in many studies. Vinshonhaler and Bass (1972) found that CAI was more effective than traditional instruction when standardized tests were used to measure performance. By applying the technique of meta-analysis to 35 studies Burns and Bozeman (1981) determined that drill and practice CAI was significantly more effective than traditional methods in mathematics instruction. Kulik, et al. (1983) also found that CAI drill and practice resulted in higher achievement. The results of research in this area tend to agree that students who use CAI drill and practice will obtain higher achievement scores than students who learn by traditional methods.

With the advent of video arcades and home computers, computer gaming has become very popular. The game format has been incorporated into educational programs by several software developers (Chaffin, Maxwell & Thompson, 1982). The research on instructional computer games is limited because the application of computer games to meaningful instruction is quite recent. In an early study on computer games, Wing (1967) found that games were at least as effective as conventional instruction. Significant gains in mathematics achievement among sixth grade students using computer games for instruction were reported by Morris (1983). On the other hand, Spivey (1985) found that there was no significant difference in achievement between groups of first graders who used computer games to learn addition and subtraction and those who did not. Bright (1985) concluded that there was no significant effect attributable to using microcomputer math games among preservice elementary teachers. However, Bright's study did not directly compare computer games with non-computer games. Dunne (1984) reported that computer games were effective instructional tools, but that the degree of effectiveness depends upon matching the style of the game with the subject matter and type of students using the game.

Research on the effectiveness of CAI has shown that in most cases students who are taught with CAI achieve more than students who are taught by traditional methods. Therefore, the value of CAI has generally been proven, but little work has been done on making comparisons across types of CAI. By comparing one type of CAI against another, it is hoped that some insight about the effectiveness of each particular type of CAI may be determined. With this knowledge, the teacher will better be able to make decisions concerning which type of CAI to use in particular classroom settings.

Statement of the Problem

The purpose of this study was to compare the effectiveness of three types of CAI lessons (informational, drill or educational game) on achievement and retention on a unit of instruction about Halley's comet. The study also examined the effect that the type of CAI lesson had on student attitudes toward CAI.

Research Questions

This study is concerned with three questions. (1) Will students in the educational game group differ significantly in achievement than either the drill group or the informational group? (2) Will students in the educational game group differ significantly in retention than students in either the drill group or the informational group? (3) Will students in the educational game group differ significantly in their attitude toward learning than students in either the dril group or the information group?

Rationale

The major difference between the groups will be in the method of CAI used to deliver the instruction. The treatments might produce difference achievement and attitude results because they vary on several characteristics including motivation, feedback, interaction and time on task as well as others. These feactors may have different effects for each of the three groups.

The group receiving only the informational CAI will probably have little to motivate it. Students will be told to read the material. No rewards will be promised or given, therefore the only source of motivation will be in the fact that they will be participating in an experiment. Students in this group will receive no feedback on their progress. They will have no way of knowing that they are learning the material--only that they have finished it. Interaction with the material will also be at a minimum. Students will press the return key when they have finished reading and will be presented with the next page of text. They will not be allowed to go back to a previous page. Students will not be given any opportunity to practice what they have read in order to aid in retention of the material. The amount of time spent on the task will probably be considerably less than the other two groups.

The motivational factor will be increased for the drill group by adding a drill after the informational lesson is read. Students will be motivated to get the correct answers to the questions, but feedback will be minimal as students will not be told whether or not their answer is correct. Interaction with the material will be increased due to the responses required during the drill. Because the drill will be given in addition to the informational lesson, the time spent on the task will most likely be longer.

Motivation will be maximized in the educational game group by the addition of an adventure-type game after the informational lesson. The game will probably be more motivating because of the goal and the graphics that will inform students of their progress toward the goal. The graphics will provide concrete feedback to students on the correctness of their answers. Because of the game element students should interact more with the material. Students in this group will probably spend more time on the task than students in either of the other two groups.

In summary, the educational game group might obtain higher scores on both the posttest and the retention posttest and have a more positive attitude toward the lesson because of increased motivation, concrete positive feedback, greater interaction, practice and more time on task.

Operational Definitions of the Variables

The following variables will be examined in this study: achievement, retention, attitude toward the learning experience and type of CAI (informational, drill or educational game). Achievement will be measured by scores on a posttest administered immediately after the completion of the unit of instruction. Retention will be measured by scores on a similar posttest administered one month after the unit of instruction is completed. The attitude of the student toward the learning experience will consist of the results of a survey of student feelings toward the type of instruction they received.

The three types of CAI that will be used in this study are informational, drill and educational game. The informational lesson will present text on the video monitor one page at a time with the student deciding when to view the next page of text. The drill lesson will be made up of two parts. Part one will consist of the informational lesson. This will be followed by questions about the informational lesson presented one at a time in flashcard format requiring students to respond by selecting the letter of the correct answer. The educational game lesson will also consist of two parts. The first part will be the informational lesson. The second part will begin with a screen explaining the rules and goals of the adventure. Next, the students will be asked to answer a multiple choice question about the material in the informational lesson. A correct answer to the question will present the student with a high resolution graphics picture on the screen giving the student feedback on his/her position relative to the goal of the adventure.

METHOD

Subjects

There were 37 sixth grade students who participated in the study. These students came from two classrooms and comprised the entire sixth grade of the school. The students were randomly assigned to one of three treatment groups--11 receiving the informational lesson, 13 receiving the drill lesson, and 9 receiving the educational game. Because of absences, four students were eliminated from the study.

Materials

The materials used in this study consisted of both hardware and software. The hardware consisted of 13 Apple II plus computers with at least one disk drive and monitor. The researcher-developed CAI lesson about Halley's Comet had three parts consisting of information, drill and practice, and an educational game which was previously described. The following researcher-developed paper and pencil test instruments were used: pretest, posttest, attitude survey and retention posttest.

Design and Procedure

The pretest-posttest control group design was used in this study. The control group was the group receiving the informational lesson only. The first experimental group received the informational lesson followed by a drill lesson. The second experimental group received the informational lesson followed by the educational game. Pretest, posttest and retention posttest data were analyzed using analysis of variance procedures (ANOVA) with alpha = .05.

On the day before the treatment, the two classroom teachers randomly assigned each student in their class a number from 1 to 37. This number was used to assign subjects to groups and facilitate data collection. Teachers then administered the pretest consisting of ten multiple choice questions.

Students in the informational lesson only group were brought into the room with the computers, read the directions and told to begin. When all students in this group had completed the informational lesson, they were given a written posttest and attitude survey.

Students in the drill group were brought into the classroom next. They were read the directions and told to begin the informational lesson. When all students in this group were finished, the informational lesson was removed from their computer and the drill lesson was booted. They were told to follow the directions on the screen and begin. When everyone was finished with the drill lesson, the posttest and attitude survey were administered.

Finally, students in the educational game group were brought into the classroom, read the directions and told to begin the informational lesson. When all students were finished, the informational lesson was removed from the disk drive and the educational game was booted up. The students were told to follow the directions on the screen and begin. The posttest and attitude survey were administered when the entire group was finished. One month after the completion of the treatment, the retention posttest was administered by the classroom teachers.

Results

The pretest means for the informational, drill and educational game groups were 3.18, 3.00, and 3.00, respectively. The standard deviations for the informational, drill and educational game groups were 1.60, 1.68 and 0.87, respectively. A one-way ANOVA done on the pretest scores detected no significant differences among the groups (see Table 1). Because the groups were statistically equivalent, an analysis of variance was used to determine if there were any significant differences among the groups on the achievement and retention posttests.

Table 1

Pretest ANOVA Summary

Source	SS	df	MS	F
Between Groups	0.2424	2	0.1212	0.06
Within Groups	65.6363	30	2.1878	-

Note. F is not significant at the .05 alpha level.

The group receiving the educational game scored highest on the posttest with a mean of 5.56 and a standard deviation of 1.87. The group receiving the informational lesson only scored next highest with a mean of 5.45 and a standard deviation of 2.66. The drill group scored lowest with a mean of 5.0 and a standard deviation of 2.77. However, the results of the ANOVA on the posttest showed no significant differences among the groups (see Table 2).

Table 2

Posttest ANOVA Summary

Source	SS	df	MS	F
Between Groups	2.0202	2	1.0101	0.16
Within Groups	190.9494	30	6.3649	-

Note. F is not significant at the .05 alpha level.

The group receiving the informational lesson scored highest on the retention posttest with a mean score of 4.67 and a standard deviation of 1.94. The group which received the educational game, scored next highest with a mean of 4.50 and a standard deviation of 1.41. The drill group scored lowest on the retention posttest with a mean of 3.82 and a standard deviation of 1.54. The retention posttest data were then analyzed using a one-way ANOVA. Again, no significant differences among groups were evident (see Table 3).

Table 3

Retention Posttest ANOVA Summary

Source	SS	df	MS	F
Between Groups	4.0779	2	2.0389	0.75
Within Groups	67.6363	25	2.7054	-

Note. F is not significant at the .05 alpha level.

An analysis of the attitude survey revealed a significant difference among the groups on the question, "I feel I have learned much about comets from this lesson." The educational game group had a significantly more positive response to this question than the information group (see Table 4).

Table 4

Attitude Survey Summary for Student Response to the Question: "I feel I have learned much about comets from this lesson."

Group	n	Mean	Standard Deviation
Informational	11	4.2727	.9045
Drill	12	3.8333	.5774
Game	10	4.8000	.4216

ANOVA Summary

Source	SS	df	MS	F
Between Groups	5.0969	2	2.5485	5.68
Within Groups	13.4485	30	0.4483	-

Note. F is significant at .05 alpha level. Game group is significantly different than drill group at .05 alpha level.

Discussion

The results of this study do not support the original hypotheses that the group receiving instruction with the educational game would score significantly higher on an achievement posttest and a retention posttest than the groups receiving instruction with an informational computer lesson or a drill and practice computer lesson. Some factors that may have contributed to these results are the sample size, the classroom arrangement and the instructional design of the computer program.

The sample size might have been too small to uncover any significant differences between the groups therefore the study should be repeated using larger group sizes. Some disruption of the normal routine of the students was caused because the entire experiment was conducted during one morning of a school day. Also, the students in the group receiving the educational game were forced to wait until the other two groups were finished to participate since the entire sixth grade was used.

Some weaknesses were discovered in the program itself that may have contributed to nonsignificant results. The readability level of the text was calculated to be high (11th grade level). The graphics were slow loading and static--possibly lowering motivation and increasing frustration for the game group. The drill was a flashcard style instead of an increasing ratio review format.

In conclusion, the results of this study provide no statistical evidence that computer games are more effective than either drills or informational type CAI programs in presenting identical content to students. There is statistical evidence that the students who received instruction with an educational game perceived themselves as learning more about the content of the lesson than students who received instruction with the informational lesson or with the drill lesson.

REFERENCES

Bright, G. W. (1985). Teaching mathematics with microcomputer instructional games. Journal of Educational Computing Research, 1(2), 203-208.

Burns, P. K. & Bozeman, W. C. (1981). Computer-assisted instruction and mathematics achievement: Is there a relationship? Educational Technology, 21, 32-39.

Chaffin, J. D., Maxwell, B. & Thompson, B. (1982). ARC-ED curriculum: The application of video game formats to educational software. Exceptional Children, 49(2), 173-178.

Dunne, J. J. (1984). Gaming approaches in educational software: An analysis of their use and effectiveness. (ERIC Document Reproduction Service No. ED 253 207)

Edwards, J., Norton, S., Taylor. S., Weiss, M. & Van Dusseldorp, R. (1975). How effective is CAI? A review of the research. Educational Leadership, 33, 147-153.

Ellis, J. D. (1984). A rationale for using computers in science education. The American Biology Teacher, 46(4), 200-206.

Kulik, J. A., Bangert, R. L. & Williams, G. W. (1983). Effects of computer-based teaching on secondary school students. Journal of Educational Psychology, 75(1), 19-26.

Lipson, J. I. (1983-1984). Promising directions for computer based instruction in science. *Journal of Educational Technology Systems*, *12*(2), 171-180.

Lunetta, V. N. & Dyrli, O. E. (1970). Computers in the science curriculum--some concrete applications in the physical sciences. *Science Education*, *54*(2), 147-154.

Morris, J. P. (1983). Microcomputers in a sixth grade classroom. *Arithmetic Teacher*, *31*(2), 22-24.

Spivey, P. M. (1985). *The effects of computer-assisted instruction on student achievement in addition and subtraction at first grade level.* Unpublished educational specialist thesis, Augusta College, Augusta, GA. (ERIC Document Reproduction Service No. ED 263 874)

Thomas, D. B. (1979). The effectiveness of computer-assisted instruction in secondary schools. *AEDS Journal*, *12*, 103-116.

Visonhaler, J. F. & Bass, R. K. (1972). A summary of ten major studies on CAI drill and practice. *Educational Technology*, *12*, 29-32.

Wing, R. L. (1967). *The production and evaluation of three computer-based economics games for the sixth grade.* (Report No. CRP-2841). Westchester County, NY: Westchester County Board of Cooperative Educational Services. (ERIC Document Reproduction Service No. ED 014 227)

Zsiray, Jr., S. W. (1983-1984). A comparison of three instructional approaches in teaching the use of the abridged reader's guide to periodical literature. *Journal of Educational Technology Systems*, *12*(3), 241-247.

Contact:

David W. McMullen
University of Illinois
Department of Secondary Education
1310 S. Sixth Street
Champaign, IL 61820

217-333-0227

ADCIS Proceedings, November 1986

Logo Tools for Teaching Algebra

Nancy Roberts - Lesley College Graduate School

ABSTRACT

Since Programming was introduced into the secondary mathematics curriculum, teachers have suggested that learning to program supports students' mathematical skills and thinking abilities. This NSF sponsored project attempts to test this hypothesis by developing introductory algebra curriculaum that connects more closely the mathematics and programming environments. The curriculum includes five modules: Variables, Expressions, and Relations; Equations and Inequalities; Functions; Story Problems; and Polynomials; and three software laboratories.

Since programming was introduced into the secondary mathematics curriculum, teachers have suggested that learning to program supports students' mathematical skills and thinking abilities. This project attempts to test this hypothesis by developing introductory algebra curricula that connects more closely the mathematics and programming environments. The 1985-86 academic year focused on materials and software development. This year's activities include pilot testing and evaluating the materials.

The curricula include five modules and Macintosh-based support software. The modules are:

1. Variables, Expressions, and Relations
2. Equations and Inequalities
3. Functions
4. Story Problems
5. Polynomials

The first four have thus far been developed. The software tools to support these modules are:

1. Marble Bag Icon Laboratory
2. Machine Icon Laboratory
3. Algebra Workbench

The software is written in Logo and Logo language programming is an integral part of the curriculum.

Three ideas are fundamental to this work:

1. Making student ideas and thinking explicit is a useful tool for developing understanding. Programming involves students in articulating what they know, externalizing the process of solving (or generating) a problem, teaching the machine, and, in some activities, anticipating how to teach other students to solve problems. Throughout the modules students write programs in Logo to generate algebraic equations or model functions.

2. New abstractions, such as those involved in learning algebra, should be tied to well-understood concrete events or to meaningful metaphors. Without this there is the danger that students learn the syntax and transformations of algebra as a kind of voodoo set of inflexible recipes. One strategy for accomplishing this is to have students experience multiple representations of an idea and learn to translate between these different representations. In this curriculum students translate among several representations of an algebraic expression using English, iconic representation, Logo programming code, and conventional algebraic notation. To facilitate the use of iconic representations the Macintosh icon labs are used.

3. Two important underlying ideas for doing algebra include:

- the composition of functions;
- the reversibility of operations.

Students are provided with instantiations of and meaningful experiences with both of these ideas throughout the modules.

One example that combines iconic representation with the idea of the reversibility of operations or *undoing* uses an iconic representation of function machines. These machines have an input hopper area where the processing occurs, and an output spout. The contents may be a familiar arithmetic function such as multiply by two or maybe a function to encrypt a message into secret code. See Figures 1 and 2.

Most algebraic expressions involve the composition of more than a single manipulation. To solve for the unknown in such expressions, a person must recognize the fundamental operations that occur and then undo each function to find the unknown. For example, the expression $3(X+5)-14$ may be seen as a three-step process in which the value represented by X is first increased by 5, $(X+5)$. The result is then tripled $3(X+5)$, and that result is diminished by 14, $3(X+5)-14$. These "steps" can be represented as function machines, as shown in Figure 2. If we are told that the resulting value is 19, $3(x+5)-14=19$, and we are to determine X, we might undo the obscuring of X's value by first "undiminishing" 19 (adding 14 to it) to get 33, then "untripling" (divide by 3) to get 11, and finally "unincreasing" by 5 (subtract 5) to get 6 as the value of X. Here we approach the problem $3(x+5)-14=19$ as a composition of functions and provide students with tools (the computerized Machine Icon Laboratory) in which they can build and manipulate machines and then solve problems in terms of "undoing", or doing the inverse.

Another software tool is programming in Logo. These functions may be written directly in Logo:

```
TO FUNC1 :NUM
OUTPUT LESS14 TRIPLE PLUS5 :NUM
END
TO LESS14 :NUM
OUTPUT :NUM - 14
END
TO TRIPLE :NUM
    OUTPUT :NUM * 3
END
TO PLUS5 :NUM
OUTPUT :NUM + 5
END
```

In more standard algebraic notation:

```
TO FUNC1 :NUM
OUTPUT 3 * ( :NUM + 5) - 14
END
```

Of course we can write Logo functions to undo each of these:

```
TO UNFUNC1 :NUM
OUTPUT :UNLESS14 UNTRIPLE
    UNPLUS5 :NUM
END
TO UNPLUS5 :NUM
    OUTPUT :NUM -5
    END
```

and so forth.

The Modules

Module 1 - Variables, Equations and Relations

The first module, informally called *From English into Algebra*, begins with students writing Logo procedures to manipulate language, in particular procedures to generate gossip. Manipulating language is a non-threatening and easy entry into algebra manipulations. Through generating gossip the imagery of machines that later serves as a model for functions is introduced.

In the first part of the module students generate examples of gossip and work together to look for a pattern. Once a pattern has been found they learn how to represent it in terms of machine imagery and then how to directly translate the pattern into a Logo program. See Figure 3.

The early focus on language leads to creating interactive quiz programs, beginning with simple arithmetic quizzes, and moving onto algebra quizzes. In addition to having a structure that supports algebra -- variables, symbolic manipulation and equations - - the quizzes afford students the opportunity to change roles from recipient to teacher and developer.

Module 2 - Number Puzzles and Marble Bags

Module 2 begins with a simple "guess my number" game (based on work originated by Max Beberman). Marble bags are the iconic images used to represent algebraic processes rather than the abstractions of X s and Y s. This "Marble Bag Laboratory" allows students to build marble bag stories and manipulate the results.

These stories can be represented in English, with marble bag icons, and with standard algebraic notation. Students are asked to go back and forth between these representations of their stories. For instance, "*Think of a number* " is called X, *add 3* results in $X+3$, *multiply by 3* is $3(X+3)$, and so forth, as illustrated in Figure 4.

Module 3

Module 3 consists of two parts: *CODES* and *GUESS MY RULE* games. In the first part, students write Logo procedures to encode and decode words and sentences, beginning with simple codes. For instance, one procedure might produce:

MPHP BMHFCSB

which is the coding of:

LOGO ALGEBRA.

The code is generated by offsetting each letter by one in the alphabet, *a* becomes *b* becomes *c*. The utility of using numbers to represent the place of letters in the alphabet is developed and students write Logo functions for encoding and decoding messages.

Students also make up their own code rules and write Logo
functions, both for coding and decoding. Undoing becomes one way to think about decoding. The codes also can be represented and constructed using the Machine Laboratory.

After working with codes based on addition and subtraction, codes based on multiplication are introduced. Encoding procedes by multiplying a letter's alphabetic number by some factor -- perhaps 2. In this case *H* (being letter number 8) becomes *P* (letter number 16). The idea of using *A* as the letter after *Z* to handle doubling that goes over 26 is introduced. The codes create function-like relationships such as $Y=X+b$ and $Y=aX$. Combining multiplication and addition in a code can become represented by an equation in the form of $Y=aX+b$. Finding the coding rule, given a coded and decoded message, is a challenge that is addressed by making graphs and doing a mathematical analysis of the results.

The second part of Module Three is *GUESS MY RULE* function games. This begins with the teacher creating charts of inputs and outputs and students having to guess what the "rule" (function) producing the output. This moves to Logo-based games where a program generates outputs based on students' test inputs. The students then guess the rule and write Logo procedures to reproduce the input and output patterns. Students also are asked to write their own function machine procedures to challenge other students. A graphing utility is part of this activity so that students can see a graph of any input/output chart to aid in figuring out a rule.

An extension of the *GUESS MY RULE* game is a sequence game where a Logo program tries to guess the rule for a number sequence input by a user. The program does not know much in the beginning, but when it cannot figure out a rule it is the student's job to teach the program the rule by writing the rule as a Logo procedure. The machine gets smarter and smarter as it is taught more rules. Students are challenged to come up with more complex sequence rules and to express them mathematically. Students can competitively construct their own machines to be smarter than any others and then how to teach the machine to use generalized rules.

<u>Module 4 - Story Problems</u>

The fourth module focuses on word or story problems starting with equations and working with students to develop story problems to fit the equations. Students then write Logo template programs to generate random story problems for a given equation. Logo function machines are then used as tools to help solve or generate word problems. Using the Machine Icon Laboratory students build simple function machines and to connect these machines together to create more complex machines, modeling the composition of functions. An example is a punch party problem:

> The sixth grade class wants to give a party. They want to know how many quarts of lemonade to make. The paper cups they already have hold 6 ounces each. They have invited 48 people to come.

When asked, "What do you need to know to solve this problem?" one student responded that you need to know how many ounces in a quart. Another said, you need to know how many glasses in a quart. A third student said, you need to figure out how many ounces will be drunk. The idea is to use Logo and the Machine Icon Laboratory to create machines that solve various parts of the problem and then put the pieces together to answer the original question. In this example, a machine to translate ounces to quarts might be the first step. Then a machine to translate people to number of ounces can be built. Each of these can be constructed and tested in the Machine Laboratory and then put together to create a *"people to quarts"* machine, Figure 5. The machines also can be turned into a Logo program:

```
TO PEOPLE.TO.QUARTS   :PEOPLE
     OUTPUT OZ.TO.QUARTS
     PEOPLE.TO.OUNCES   :PEOPLE
END
```

Once the machines are built they can be used to solve other similar problems. Given a set of guided experiences with several of these problems, students find it easier to take other similar problems apart. Having students approach problems by building simple machines helps them isolate different aspects of problems and aids them in parsing problems into useful pieces. Interpretation and extension of each problem results as the class tries to construct solutions. In the lemonade example someone might ask, "What if the people who come to the party drink more than one glass of lemonade?" (a June party in Atlanta). This leads to the question, "Can we fix our machine so it can handle this possibility? What do we have to change if each person drank two glasses?"

A further step in creating these machines is to ask students if they can build more general purpose machines, so they could handle different sized cups, or different numbers of cups per person in a new lemonade problem. By replacing constants with variables students can create more general purpose tool programs. The machines students create can become tools in a toolkit for exploring (and generating) related word problems. This opens the way for the creation of sets of word problem-solving tools centered around different domains.

The final piece of tool software is a symbol manipulation workbench that helps students apply the ideas they have developed to manipulating expressions in algebraic notation. The Algebra Workbench focuses on the lower level skills of dealing with and manipulating standard algebra notation.

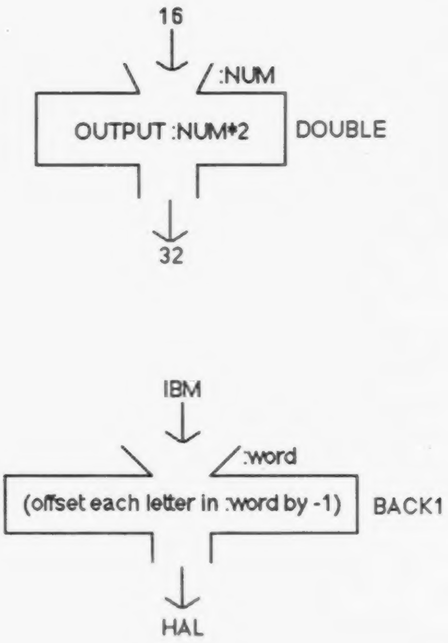

16
:NUM
OUTPUT :NUM*2
DOUBLE
32
IBM
:word
(offset each letter in :word by -1)
BACK1
HAL

Figure 1. A Multiply by Two and an Encrypt Machine

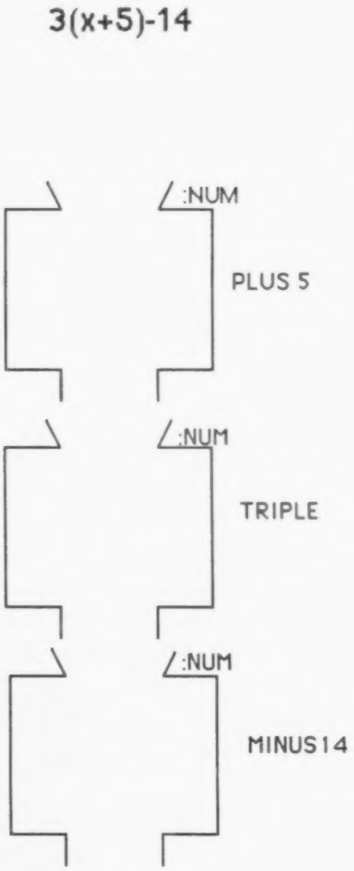
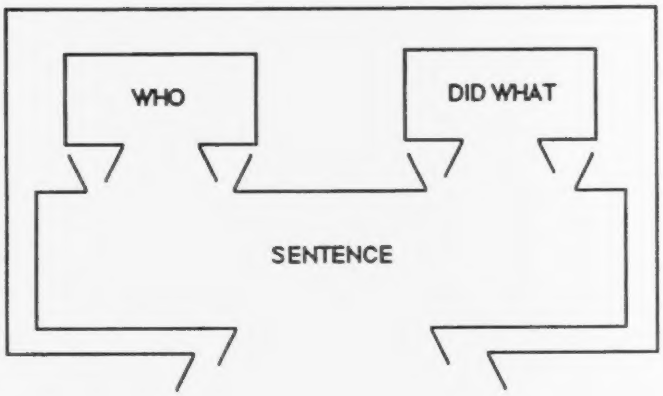

3(x+5)-14
:NUM
PLUS 5
:NUM
TRIPLE
:NUM
MINUS14

Figure 2. Solving for X Using the Machine Icon Laboratory

A GOSSIP MACHINE


WHO
DID WHAT
SENTENCE

```
TO GOSSIP
OUTPUT (SENTENCE WHO DID.WHAT)
END
```

Figure 3. Using the Machine Icon Laboratory to Genrate Gossip

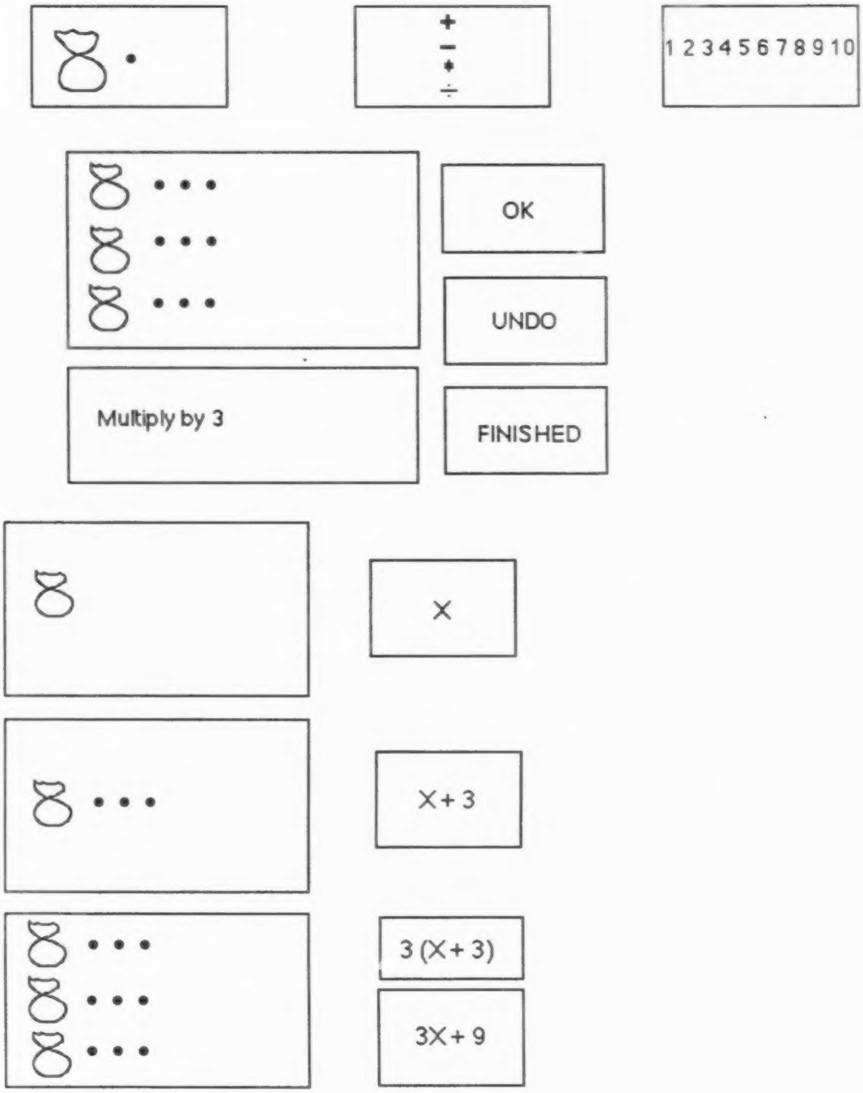
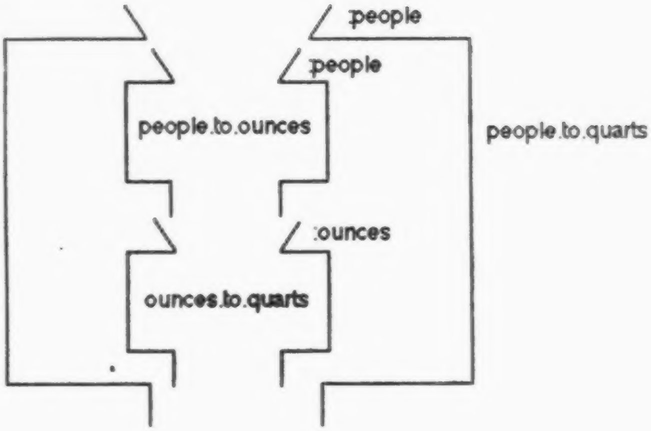

+
−
*
÷
1 2 3 4 5 6 7 8 9 10
OK
UNDO
Multiply by 3
FINISHED
X
X+3
3(X+3)
3X+9

Figure 4. The Marble Bag Icon Laboratory


:people
:people
people.to.ounces
people.to.quarts
:ounces
ounces.to.quarts

$$\frac{(P \cdot 2) \cdot 6}{32} = \text{QUARTS}$$

Figure 5. People to Quarts Machine

ADCIS Proceedings, November 1986

EFFECTIVE ADMINISTRATION OF THE MICROCOMPUTER LAB IN HIGHER EDUCATION

DR. CHARLOTTE SCHERER

BOWLING GREEN STATE UNIVERSITY

ABSTRACT

Several factors need to be considered when establishing and administrating a computer lab in a higher education institution: purpose, space, furniture, hardware purchase, software purchase, staff, budget, growth and expansion, and operating policies and procedures. Each of these factors has a number of complications which are presented and explained. Possible solutions to the problems are discussed.

INTRODUCTION

The growing need in institutions of higher education to establish microcomputer labs for student and faculty use entails the responsibility for administrating these labs in an efficient and cost-effective manner. A number of factors make this task more complex than appears on the surface. This paper will attempt to list and discuss these factors with the intent of describing some of the problems that arise and suggesting means for coping with them.

ESTABLISHING THE NEW LAB: GENERAL CONSIDERATIONS

In order to establish a new computer lab somewhere on campus, a number of things need to be considered: purpose, location, furniture, hardware purchase, software purchase, staffing, budget, growth and expansion, operational procedures and policies regarding use of the lab.

Before anything else can be considered, the purpose(s) for the lab must be established. Is it for a particular group of students? a particular department or program? Is the lab for classroom instruction or will it be an open lab to support classroom instruction that takes place elsewhere? Or is it to serve dual purposes? How many classes and faculty is the lab supporting? What will students and faculty do in the lab? Programming?CAI? Wordprocessing?Other applications?Where is the lab to be located? In a residence hall? a revamped classroom? a room in the student union? Once these questions are answered, it will be much easier to decide on the appropriate hardware and furniture to furnish the lab.

The design of the lab depends on the layout of the room available and purposes for which the lab is used. Approximately 35 to 50 square feet per person should be allotted for each station. If the lab is to be an open lab, the arrangement of furniture should provide as much workspace as possible near each computer where users can put materials with which they are working. If programming is the main consideration, then work tables near the computer stations allow for spreading out written work during debugging sessions and free up the machines for others to use. If the stress is on individual work in a quiet environment, carrels especially made for computers could be a real asset, but will cost more than long tables at which three or four computer stations can be placed. The number and kind of printers depends upon the purpose(s) for the lab, but a ratio of one to every four or five computers is probably the minimal requirement in either an open lab or an instructional lab.

If the lab is to be used primarily for instructional purposes, some workspace near each station is important, but more important is an arrangement of furniture such that an instructor can demonstrate from a workstation in the front, but get around easily to check how students are doing at their own workstations. A projection system is a great asset to this instructional arrangement, or a video controller or hard diskdrive networking system which allows the instructor to control the various machines in the room could be the choice. Making this decision depends partly on the hardware chosen and partly on the nature of the software and the licensing or copyright policies associated with it. Some software cannot be legally networked, but could be used with a video control system, provided the monitors are compatible.

There are other important features to consider when furnishing any lab. Table or carrel height should be 26.5 inches from the floor and chairs should be adjustable to allow appropriate distance from the keyboard. Adequate lighting, free from glare on monitors, even temperatures of the room, air conditioning during warm seasons and proper air circulation in a crowded room, also are essentials if users are to have adequate view of the screens, users and staff are going to have a comfortable work and learning environment and machinery is to be kept in good condition. An alarm system is another important consideration for security purposes. The lab facility needs to be alarmed at the doors and a minimum of keys allotted to those monitoring use. Individual machines can also be bolted to tables or carrels if this seems necessary. A secure system for circulation of the software will be discussed in the section on Staffing.

Hardware Purchase

The type of hardware to be purchased depends upon the overall purposes of the lab and the software chosen to meet those purposes. The two cannot be decided independently. If, as in our College of Education and Allied Professions lab, the main purposes are instruction in educational computing and teaching basic administrative tool applications such as integrated packages of wordprocessing, databases and spreadsheets, then the likely hardware becomes Apples and IBM's; Apples because they are so heavily used in K-12 school settings and IBM's because they are more likely found in administrative office settings. Because color is so important to educational software, color with good resolution or, better yet, RGB monitors are more useful than monochrome. Inexpensive color monitors with poor resolution will not work effectively with 80 column boards for wordprocessing. If tight budgets prohibit the more expensive color or RGB monitors, then a mix of monochrome and color is better than no color and provides flexibility for tool uses. If programming, using a mainframe, is the key purpose, then standalones that can act as terminals for the mainframe or a mixture of standalones and those through which programs can be submitted to the mainframe seems to be the best solution. The key guideline to hardware purchase is the chief purpose for which the lab is to be used. The more multipurpose the lab, the more difficult the hardware decisions. For ease of learning where the major use is for tool applications, especially wordprocessing, the Apple Macintosh might well be the appropriate choice.

Trying to plan hardware for five years down the road is almost impossible. One lab developed under a three year plan ended up buying different machines each year, because the hardware had been upgraded by the time of the next purchase. This isn't all bad, but it does present about a 90-95% compatibility problem with software that is also being upgraded rapidly. Nor does this situation appear likely to improve in the near future. According to a recent article in Education Week(Snider, 1986) no real standardization of machinery by the various companies is in sight in the next several years. Any machinery presently purchased will be obsolete in less than five years; however, choosing standard machinery that can be enhanced and expanded or eventually replaced seems to be the best choice and some companies are offering upgrade programs that make this feasible.

Another strategy suggested by Snider is to bypass micros entirely by choosing a mainframe or mini with separate terminals. However, this strategy works only if the software needed is available for the mini or mainframe and provided that one of the purposes is not to familiarize the student with the micro that is apt to be found in the workworld. The choice of other peripherals, such as printers, modems, extra drives, and video, again depends upon the purposes for the lab. If wordprocessing is the key purpose, then more printers will be needed than for CAI classes, or for viewing sample K-12 software.

Generally, printers can be set up to serve two different machines by switching a black box from Machine A to B when ready to print. This saves having to pull disks out and take them to a machine with a printer, but also means that more printers must be purchased. For most purposes, high quality dot matrix printers with ribbons changed fairly frequently can be quite adequate for most student needs; however a good daisy wheel letter quality printer or one of the new laser printers should be available somewhere on campus. Our university has dot matrix for everyone, daisy wheels available at no charge to faculty and graduate students, and a laser printer available to anyone at minimal charges.

Software Purchase

The purchase of the appropriate software to meet the needs of the lab users and to work on the machines is not too difficult with just one relatively single purpose lab, but it can become very complicated as new labs are established and there is need for compatibility between the various labs. Because vendors are not standardizing many products and often upgrade software as fast as they upgrade machines, the version that works on the older machine is often no longer available, or even if it is, it does not work on the newer machines. In a large university with multiple users who have learned to use particular software, which may not be available in all labs, this causes inconvenience and contributes to the problem of piracy.

The problem of incompatibility of upgraded software with older machines or vice versa remains a real one, and there are no easy solutions. One case in point is Applewriter, a word processing program for the Apple II series. To date, this author has learned and used four different versions of this program--two designed for the Apple II+, one for the older Apple IIe(which is not fully compatible with an enhanced IIe) and finally, the ProDOS version (which requires an 80 column board to work effectively). Apple has agreements with the author which do not allow the Applewriter II version to be put in the public domain, yet it is no longer being sold. The only solution we've found is to try to relegate our older machines to labs using mostly CAI which does not require large memory or ProDOS operating systems, and to try to convince our users either to learn the new version or try to work only in the labs which have the older machines. This is inconvenient and does not make users happy, but it is the state of the art and it becomes necessary to resign oneself to certain facts of life in an age of technology.

Another general problem with purchasing software involves staying within copyright law. Recently, the copyright law has been interpreted so as no longer to be a gray area with respect to multiple loading of software in several machines (Finkel, 1985). It is generally considered illegal to multiple load most software in several machines. However, different software vendors have established different policies in regard to multiple loading, networking, onsite licensing, and lab pack discounts: thus, the purchaser must learn the individual company's policy in order to make intelligent decisions about how many copies to purchase. The more positive side of this is that many vendors are offering quantity purchase discounts or lab pack purchases and are now advertising that fact, so that it is easier today than it was last year to acquire multiple copies at lower prices. Also, many vendors are granting permission to educational institutions to multiple load if such permission is requested from the company. It is also generally agreed that if the lab staff puts up signs reminding users of copyright law and informing them that pirating of software is illegal, they are protecting themselves against possible law suits when users use lab equipment to make illegal copies. Of course, a policy regarding punishmnet for apprehending offenders should also be in effect and publicized to all users.

Another potential problem with copyright law and software purchase involves the use of hard disk networks. Some vendors do not allow software to be put on such a network unless it is a specially licensed piece of software for network use only and then it is considerably more costly than it would be otherwise. Often, too, the software desired does not come in a network version. One possible solution to this problem would be to consider a video control system, rather than a hard disk network, since this wouldn't involve loading software onto a hard drive, but would only switch the video pictures on the various screens in the lab or classroom.

A further complication involved with software purchase is also related to budgeting problems. In our case, there are three College of Education and Allied Professions Microcomputer labs and classrooms all under one administrator who coordinates the use, establishes policy and procedures, makes purchases, hires and trains monitors, and generally oversees the smooth functioning of each facility. The problem arises out of a policy that originally had the College supporting hardware purchase and departments purchasing software that is to be used in common in the lab or classroom/lab. Such an arrangement raises the question of who is responsible for the software. Who replaces it when it is damaged by a user in another department? The solution we have found is to change the policy and have a central budget for purchase, maintenance and replacement, when necessary, of most common software for the labs. It then becomes necessary for the lab administrator to check with, or hold meetings with representatives from the various departments to get consent on purchase of software in common. The more esoteric software items can be purchased by individual departments and they can then be the sole users of that software. This solution has proved effective thus far and seems to cause the least problems in dealing with software purchase. The important point here is that software purchase, maintenance and upgrade is costly and needs to be a line item in the lab budget from the very start. Too often when a lab is being established, hardware purchase is accounted for, while software purchase is not seen as the essential item it soon becomes if the lab is to function effectively. All software purchased should be processed and catalogued by lab staff.

STAFFING

Once the lab has been established with purpose, furniture, hardware and software, the next consideration is staffing. If it is an instructional lab only, the instructors can monitor the use by students of both the machines and software. Though this seems an acceptable way to do things, whether this will work well depends upon the individual faculty members. Faculty must organize and administer systems for checking out software during class time, for making sure machines are shut off and the room locked at the end of the session with keys returned to the appropriate place. This doesn't always happen, but most faculty try to be cooperative. For any open lab hours, it is essential to have staff monitors to circulate software and generally oversee the functioning of the lab. Student monitors can be hired and trained for these duties, thus keeping costs to a minimum. A more permanent clerical assistant is very helpful for processing all software as it comes in, keeping records and counts, scheduling hours for student monitors to work and processing their pay forms.

The training the monitors need depends upon the purpose for the lab. In a lab where much programming is done, monitors who can be consultants to novice programmers are very useful. In a lab where CAI is done, monitors should be familiar with the machines and the CAI programs being used, but need not know in great detail what a particular program is about. In our College of Education and Allied Professions facility where students are learning tool software such as wordprocessing, databases and spreadsheets, we hire students who have had at least one computer science course and thus are familiar with computer operation in general. We train them in the use of the tool programs and any other special programs a teacher education major might use. We also teach them the circulation system (computerized in one lab, cards in another) and go over user policies and procedures with them. This training time takes about 6 or 8 hours at the beginning of the semester. Personal conversations, notes, phone messages, and periodic newsletters keep their knowledge current during the remainder of their employment. We try to hire sophomores or juniors who are likely to work several semesters before graduation, thus reducing turnover. Experienced lab monitors also help with the training of new ones and we pair them up after a certain amount of formal instruction by the lab Administrator and her assistant. Lab monitors are to get their own substitutes if possible so generally the lab is covered all the necessary hours, with very few problems of absenteeism or tardiness.

The circulation system in one of our larger labs is computerized using a hard drive and database program especially written to serve our needs. User ID's are input at the beginning of the semester as much as possible and reports can be generated from the system. Student monitors are taught to use the system and to input User ID's, which can be done during less busy times. In our smaller lab where we are on a card circulation system, the student monitors make up new cards when necessary and do manual counts of usage during less busy times. All users are required to leave an ID in exchange for software and the ID is returned when the software is checked back in. This has virtually eliminated the problem of "walking" software.

BUDGET

With regard to budget, once the initial expenditures of equipping the facility are considered, the main operational costs are personnel, clerical supplies, maintenance of equipment and as mentioned previously, continual purchase of software, to stay as current as possible. Expenditures for all of these items depend upon what is available at the particular institution, the size of the lab, the amount of usage and what the purposes of the lab are. Institutions such as ours, with large Computer Services offices, can usually get free labor maintenance with costs only for parts. If this is not the situation a service contract with the vendor should be considered. Student personnel is usually less expensive than classified permanent clerical help, but the time spent in turnover and training partially offsets the savings. Software update and purchase of new equipment depends partially upon the purposes of the lab, whether or not the student population is growing and how heavy the usage is in terms of wear and tear on the machines. Equipment not used more than eight or ten hours a day will last longer than equipment used heavily up to 24 hours a day as is the case in some computer science labs. Generally, after three years of heavy usage, some machines tend to start needing major repairs which must be accounted for in a maintenance budget. This is not to say that some querky machines may not need repairs even after three or four months, but this is unusual and often the repair is inexpensive.

OPERATIONAL POLICIES AND PROCEDURES

For efficient administration of the lab, it is necessary to devise a statement of policy and guidelines for use. In our College, a set of guidelines for use by faculty, students, and staff is published and sent to all faculty and staff. It is also explained to student monitors during their training, and a copy of these guidelines is available in each lab. Brochures distinguishing the purposes, available equipment and use of each lab are also available in each lab and distributed to all faculty and staff within the College, as are bookmarks which state the hours the lab is available.

The instructional/open lab is scheduled first for classes, then for open hours of usage. During open hours of use, the lab may be booked in advance for a one or two hour period for a group to see a demo, learn a particular program or whatever. Such bookings are posted on a board near the facility. The remaining open lab hours are on a first come, first served basis with no reservations taken. During peak usage, machines with printers are to be used last by those not printing, thus keeping these generally free for those who need to print. We maintain a priority for College of Education and Allied Professions users, but do not exclude others from the university when machines are free. A machine on a cart is available in one of the labs to go down to a classroom for demonstration purposes, but must be booked in advance. The other open lab maintains hours not always available in the instructional/open lab; thus it is possible to have more total hours of open lab available to the various users. The third lab, which is for instructional purposes only, is booked centrally for group purposes other than scheduled classes. There is a fee structure for groups outside the college to rent the labs on weekends or during certain less used times for workshops or presentations. These fees, though not large, contribute to the ongoing costs of running and maintaining the labs. Renting some equipment to faculty or graduate students during the less heavy use of the summer is under consideration. Other computer labs on campus do this and it too makes good use of otherwise idle machines and brings in some extra funds without a great deal of risk involved.

CONCLUSION

From the above, it can be seen that good planning is absolutely essential when the lab is being established. Its purpose(s) must be well defined and in light of the purpose(s), the various other factors of space, furniture, hardware and software purchase, staffing, budget, and operational policies and procedures need to be considered carefully. Only then can a microcomputer lab in a higher education institution be efficiently administrated in a cost-effective manner.

REFERENCES

Finkel, L. (1985). Software Copyright Interpretation. The Computing Teacher, 12(6), 10.

Ramsay, P. & Randol, L. (1986, March). Running a University Micro Lab: Ideas & Problems. Paper presented at the meeting of the Michigan Association for Computer Users in Learning, Grand Rapids, MI.

Snider, W. (1986, February 19). Mixed Signals: Computer Plans Frustrated by Incompatibility. Education Week, pp. 1,19.

Contact: Charlotte Scherer
Clinical and Computer Labs
College of Education and
Allied Professions
215 Education Building
Bowling Green State University
Bowling Green, OH 43403-0243

419-372-7392

ADCIS Proceedings, November 1986

Gender Similarities and Differences in Sixth Grader's Attitudes Towards Computers: An Exploratory Study

Marc Swadener, University of Colorado-Boulder

Michael Hannafin, Pennsylvania State University

Abstract

The attitudes of sixth grade girls and boys towards the computer were examined. An attitude instrument, consisting of three subscales and three individual items related to sex-role bias, was administered to 32 students. Responses were analyzed as a function of sex of student and entry mathematics achievement. The results indicated significant differences between the self-confidence expressed by high achievement versus low achievement students. In addition, marginal interactions were found for responses to the perceived utility and general attitude subscales, with high achievement females and low achievement males responding comparably, while low achievement females and high achievement males responded comparably. Significantly, no differences attributed to gender were found on items pertaining to sex-bias for computers.

Introduction

Past research with computers has shown that attitudes affect interests (Hannafin & Cole, 1983). In addition much has been written about the societal affects of changing technology (Lias, 1982; Logsdon, 1980). The demands of these technological changes on the individual with respect to occupational opportunities, the effect of inflation on purchasing power, rising housing costs, increasing numbers of single parents and other social factors are leading to greater concern for maximizing future opportunities for school children.

One major concern in this endeavor is to assure that females are encouraged in areas such as technology and computers, which traditionally have been dominated by males (McClain, 1983). According to some writers, persons with limited or nonexistent computer skills will face bleak futures in the job market (Evans, 1979). It is therefore important to have in depth knowledge of the differences between the sexes in attitudes towards technology and computers, because these attitudes will to some degree dictate interests. If attitudinal differences between the sexes are identified there exists a greater possibility of remediating these differences. In addition, if differences in attitude exist as a function of learner achievement in related areas, and if these differences favor high performing students, there should be concern for future employment options of the low students. Corrective action within the school program may alleviate such differences.

In this project, an attempt was made to ascertain differences, if any, that existed between sixth grade students of high and low mathematics achievement with respect to attitudes toward computers and their use. What little is known regarding such attitudes has been derived from older populations, such as high school and college students. Mathematics performance, for example has been highly correlated with computer attitude (Hannafin and Cole, 1983) and aptitudes with older students (DeBlassio and Ball, 1981). However virtually no comparable research has been conducted with younger learners. It seems plausible that the socio-cultural influences that are believed to influence sex related differences in older populations may not yet be present in younger, elementary children. A major effort within the project, therefore, was to develop an instrument which was sensitive to such differences at the sixth grade level.

Methods

Subjects

Subjects for the study were 32 randomly selected sixth grade students. They were selected from two schools in a moderate sized middle-class suburban/rural school district. The community has a combined Spanish origin and non-white minority population of about 15% of the population.

Instrument

The instrument used in the study was adapted primarily from the Computer Literacy and Awareness Assessment Instrument distributed through the Minnesota Educational Computing Consortium (MECC) (Anderson, Krohn & Sandman, 1980). The primary adaptations pertained to the reading level and base of experience for sixth-graders regarding computers, since the MECC instrument is designed for secondary age students in the unaltered form. The adapted version, consisting of 17 Likert-type items with a five part scale format ranging from "Strongly Disagree" to "Strongly Agree," was developed for local use during the project.

The initial draft of the instrument consisted of 30 items keyed to four computer attitude categories: self-confidence in the use of computers, perceived utility of computers, general attitude toward computers, and sex roles in computers. The thirty items were administered to 61 sixth graders and their responses were analyzed to determine item discriminability within the scales, and to provide measures of reliability of the four scales. Items were retained within each of the four subscales only if the individual item-subscale correlation was greater than .40. This procedure was repeated in order to discard items which were not consistent with the construct assessed by the individual scales. The procedures were repeated until individual scale reliability was a minimum of .60. The final 17 items, shown with item-scale keying, are shown in Table 1. During administration of the final instrument, however, the items were ordered randomly and without the labeling of sub-scales. Overall and sub-scale Hoyt reliability estimates for the 17 item final instrument are given in Table 2.

Table 1
Computer attitude scale with % of cases responding to each alternative.

	% of N for response				
	SD	D	NO	A	SA
SELF-CONFIDENCE					
1. I am happy when I can make a computer do what I want.	0.0	0.0	18.8	56.3	25.0
2. I am good at running a computer.	6.3	9.4	53.1	28.1	3.1
3. I don't work well with computers.	6.3	15.6	31.3	31.3	15.6
4. I learn a lot with a computer.	3.1	3.1	18.8	43.8	31.3
COMPUTER UTILITY					
5. People don't need computers.	3.1	3.1	21.9	40.6	31.3
6. Using a computer is a waste of time.	0.0	0.0	15.6	37.5	46.9
7. Computers are a tool just like a hammer or saw.	9.4	21.9	28.1	31.3	9.9
8. Being able to work with a computer is important.	3.1	3.1	12.5	62.5	18.8
9. I would like to have my own computer.	0.0	9.4	3.1	21.9	65.5

(Continued on the next page.)

Table 1 (Continued)
Computer attitude scale with % of cases responding to each alternative.

	% of N for response				
	SD	D	NO	A	SA
GENERAL ATTITUDE					
10. I enjoy using the computer.	0.0	0.0	25.0	25.0	50.0
11. Working with the computer makes me want to learn.	3.1	3.1	15.6	56.3	21.9
12. I don't like working on a computer.	6.3	0.0	18.8	37.5	37.5
13. I like using computers in school.	0.0	3.1	28.1	28.1	40.6
14. I like to use a computer.	3.1	0.0	18.8	21.9	56.3
SEX BIAS (Used as individual items, not as a subscale.)					
15. Using computers is more for boys than for girls.	3.1	0.0	15.6	15.6	65.6
16. Computers are more for boys than girls.	0.0	3.1	21.9	31.3	43.8
17. Girls do just as well as boys in computer jobs.	3.1	6.3	6.3	37.5	46.9

NOTES: Items shown by subscale for illustration purposes only; actual scale included items in random order and without subscale indicators. Scale and weights are:
SD = Strongly Disagree (0), D = Disagree (1),
NO = No Opinion (2), A = Agree (3), SA = Strongly Agree (4).

Table 2
Instrument and subscale reliabilities.

SCALE	RELIABILITY
Self-Confidence	.68
Computer Utility	.69
General Utility	.74
Composite	.84

NOTE: Items within the sex bias category were analyzed on an item-by-item basis, not as a subscale.

Procedures

The students were divided into two groups: sex of student, and high and low mathematics achievement. Mathematics performance was determined based upon the level of achievement among the students on a district-wide criterion referenced mathematics skill test. The rationale for including mathematics achievement as a design variable was based upon the previously reported strong relationship between mathematics achievement and computer performance (DeBlassio & Bell, 1981; Fowler & Glorfeld, 1981). The generalizability of this research to younger students was of interest. Preliminary analyses of the mean scores on the criterion-referenced test indicated no mathematics achievement differences between males and females in the sample.

Testing of student attitude toward computers was done at the end of the school year. There was no formal in-school use of microcomputers by students in either of the schools prior to this study.

All subjects were administered the computer attitude instrument in their classrooms. The responses were subsequently keyed to the appropriate scales, and the composite scale scores were analyzed for differences.

Results and Discussion

The mean scores for each scale are contained in Table 3. Effects were

Table 3
Mean Item Score for Each Attitude Sub-Scale by Group.

	Mathematics Skill Level								
	Low			High			Combined		
Scale	Mean	S.D.	n	Mean	S.D.	n	Mean	S.D.	n
Self-Confidence									
Female	2.65	0.70	5	2.73	0.65	10	2.70	0.64	15
Male	2.31	0.62	12	3.15	0.22	5	2.56	0.66	17
Combined	2.41	0.64	17	2.87	0.57	15	2.63	0.64	32
Computer Utility									
Female	3.28	0.63	5	2.92	0.68	10	3.04	0.66	15
Male	2.68	0.69	12	3.24	0.17	5	2.85	0.63	17
Combined	2.86	0.71	17	3.03	0.57	15	2.94	0.64	32
General Attitude									
Female	3.28	0.63	5	3.12	0.63	10	3.17	0.61	15
Male	2.77	0.78	12	3.68	0.30	5	3.04	0.79	17
Combined	2.92	0.76	17	3.31	0.60	15	3.10	0.70	32

Note: Table includes only students for whom complete test and attitude data were available.

analyzed using ANOVA procedures for each scale in order to more clearly identify effect sources. Items pertaining to sex bias toward computers were also analyzed separately using univariate ANOVA procedures. Due to the exploratory nature of the study and the relatively small sample size available for the study, liberal p values have been reported and the findings interpreted accordingly. The results of the analyses are summarized by scale.

Self-Confidence

A significant effect was detected for mathematics achievement, $F(1,28) = 4.10$, $p < .05$. This effect was characterized by high achievement students expressing greater agreement with statements related to confidence in their computer abilities than low students. Further analysis of this scale indicated virtually no differences in the responses of male versus female students, and no interaction.

Perceived Utility

A marginal interaction was found between sex of student and mathematics achievement, $F(1,28) = 3.69$, $p < .07$. This effect was characterized by low achievement females agreeing that computers have substantial utility and high females rating the utility of computers somewhat lower. The pattern was reversed for males, with high students indicating greater utility of computers than low students. However, no main effects were found for either sex of student or mathematics achievement. Additionally, all mean response scores for this scale were essentially proximal to the "no opinion" category, and the mean score differences were relatively small.

General Attitude

Similar to the findings for perceived utility of computers, a sex of student-by-mathematics achievement interaction was obtained, $F(1,28) = 4.57$, $p < .04$. The pattern of the responses was identical to the interaction obtained for perceived utility, with high females and low males reporting less agreement, and low females and high males reporting greater agreement, with positive statements towards computers. Again, no main effects were found for either sex of student or for mathematics achievement.

Sex Bias Questions

Consistent with the overall findings, the three questions focusing on females versus males in perceived aptitude for, and probability of success with, computers, yielded no differences. Based upon the findings for computer attitudes for high school students (Hannafin & Cole, 1983), it was tentatively hypothesized that each sex would rate their gender more favorably in both regards, or that tacit sociocultural influences would result in relatively higher ratings for males by both sexes. These hypotheses were clearly disproven. In all cases, the ratings by male and female sixth graders towards both their own and the opposite sex were virtually identical. No differences were found for ratings of statements suggesting that computers are more for boys than for girls, computers are more for girls than for boys, or boys and girls will do equally well in computer jobs.

General Discussion

The purpose of this study was to examine the similarities and differences in computer-related attitudes between sixth grade boys and girls of different mathematics achievement levels. The findings tentatively suggest that boys and girls at this level respond with far greater similarity than difference, but that self-confidence may be an important factor computer participation for both sexes.

Several outcomes of this study warrant additional discussion. First, the absence of sex-related main effects may support the notion that students at this age simply do not possess the same biases toward the opposite sex as do older students. It is encouraging to obtain evidence that biasing effects have not yet affected the perceptions of sixth graders. While it is possible, indeed likely, that biasing influences exist at all ages, they do not appear to manifest themselves significantly until later grade levels. In contrast, however, it is alarming to learn of pronounced bias effects among older learners, such as high school and college-age students (see, for example, DeBlassio & Bell, 1981). Although we must continue to promote the notions of equality at the elementary level, perhaps the more pressing immediate priority exists at the secondary school and college levels.

The interactions found for both perceived computer utility and for general attitude are perplexing. High mathematics achievement females responded comparably with low achievement males, while low females responded similarly with high males. This finding was not anticipated in this study. Inspection of the mean responses for the perceived utility an the general attitude scales indicated that the largest differences among the various means was never greater than one scale point, and that all mean scores were centered about the middle scale value, "no opinion." This may indicate that the differences, though reliable, are of little practical consequence. The nature and magnitude of the response differences are simply too tentative to interpret as "real" differences at this time.

The greater confidence reported by the high versus low mathematics achievement students is consistent with the hypothesis that perceptions of confidence moderate, to a great extent, student attitudes toward the computer (Hannafin & Cole, 1983). Unlike the findings reported for high school students, however, it is encouraging to find that sex-related influences did not moderate this effect. The number and type of opportunities afforded by the "computerization" of our youth should be devoid of gender-specific expectations. It appears that many elementary school students of both sexes view the opportunities as the domain of the most able irrespective of gender.

Although gender-based differences in confidence were not found, several issues regarding the development and modification of such attitudes require attention. Students of both sexes develop attitudes towards computers through a combination of influences including both socio-cultural factors and direct experience. Attitudes are often cultivated on the basis of misperceptions of difficulty or the appropriateness of the computer for assumed student stereotypes. Bright and high aptitude students, for example, are more likely to participate in computer-related coursework than average or medium-to-low aptitude students. This may be due in part to the misperception that computers are the exclusive domain of the academically superior student. At the high school level, differences between the attitudes, and participation levels of boys and girls emerge (Research Bulletin, 1983). It seems likely that the same types of misperceptions affecting confidence contribute to these evolving gender differences.

It also appears that attitudes based on misperceptions dictate the opportunity for first-hand experience with computers by discouraging many students from participation (Cole & Hannafin, 1983). Researchers have cautioned school personnel to account systematically and programatically for misperceptions of the difficulty of the computer experience. While student confidence in the ability to use computers will inevitably continue to be greatest for those with the greatest aptitude, the low confidence student of either sex must still be encouraged to participate, to learn, and to become desensitized to their apprehensions about computers.

The relationship between mathematics achievement and computer aptitude may be less salient for younger students than for older students. In the present study, the rationale for selection of mathematics achievement as a tentative predictor of computer aptitude was based on strong correlations obtained using high school and college aged populations. In fact, analysis, subsequent to the present study, of data from a more extensive sample of sixth grade students has shown a correlation of .28 ($p = .02$, $n = 58$) between mathematics achievement and computer attitude. Perhaps mathematics at the elementary school is too computational in nature to be an effective predictor of the higher order problem solving associated with computer aptitude. In effect more powerful control variables may exist at the elementary level, than computational mathematics skill tests, in predicting computer aptitude and examining computer attitude differences of elementary students.

Students with limited computer experience may have experienced difficulty in responding to some items included in the attitude scale. Since no formal in-school experiences with computers were provided prior to the study, only students with informal, home or extra curricular computer experience could reply based upon first-hand experience. The inexperienced student in effect, may have "estimated" a response to certain items based upon individual beliefs rather than active experience. If so, this could explain the concentration of "no opinion" responses to items requiring ratings based upon experience.

A number of precautions should be exercised when interpreting the findings of this study. First, the study was exploratory and highly applied in nature. From a pure experimental viewpoint, the study was not designed to be a rigorous test of established hypotheses. The relatively limited power of the design resulting from the small sample size also may have hindered the ability to detect small, but reliable differences in student attitudes. If so, perhaps replication with larger samples is indicated. In addition, the focus on sixth grade students was important as an exploration of hypothesized male-female achievement pattern differences in mathematics and science. However, inference beyond this age-grade level--especially for higher grades-- cannot be made. It seems more likely that the results of this study are generalizable to lower grades, although even this assumption requires empirical confirmation.

The results of this study tentatively confirm the similarities in the attitudes and perceptions of sixth graders toward computers. Given this outcome, the reported differences in attitudes for older students assumes greater importance and urgency. Given the findings of this study, contrasted with previous research for older students, it appears that significant alterations in the attitudes of both boys and girls occur during the relatively brief three-to-six year period after elementary school. Perhaps attempts to understand more precisely the influences affecting perceptions toward computers in post-elementary settings will yield practical approaches to the design of needed interventions.

>>>>>> <<<<<<

References:

Anderson, R.E., Krohn, K., & Sandman R. (1980) User's Guide for the Minnesota Computer Literacy and Awareness Assessment. Minnesot Educational Computing Consortium, St. Paul, Minn.

Cole, D. D., & Hannafin, M. J. (1983) An analysis of why students select introductory high school computer coursework. Educational Technology, 23(4), 26-29.

DeBlassio, J. K., & Bell, F. H. (1981) Attitudes towards computers in high school mathematics courses. International Journal of Mathematical Education in Science and Technology, 12, 47-56.

Evans, C. (1979) The Micro Millenium. Viking Press, New York.

Fowler, G. C., & Glorfeld, L. W. (1981) Predicting aptitude in introductory computing: A classification model. AEDS Journal, 14, 96-109.
Hannafin, M. J., & Cole, D. D. (1983) A comparison of factors affecting the elective selection of introductory computer courses. AEDS Journal, 16, 218-227.
Lias, E. J. (1982) Future Mind. Little, Brown and Company, Boston, Mass.
Logsdon, T. (1980) Computers and Social Controversy. Computer Science Press, Rockville, Maryland.
McClain, E. J. (1983) Do women resist computers? Popular Computing, January, 68-70.
Nelson, L. R. (1974) Guide to LERTAP Use and Interpretation. Department of Education, University of Otago, New Zealand.
Research Bulletin (1983) Sex Differences - Computer Training: Why Do Boys Outnumber Girls? Phi Delta Kappa Center on Evaluation, Development, and Research, Bloomington, In. (May Issue).

Contact: Marc Swadener
School of Education
Campus Box 248
University of Colorado
Boulder, CO 80309
303-492-6555

ADCIS Proceedings, November 1986

THE RELATIONSHIP BETWEEN STUDENTS' PRIOR PROGRAMMING EXPERIENCE AND THEIR PERFORMANCE IN AN INTRODUCTORTY PASCAL PROGRAMMING COURSE

Michael L. Waugh

University of Illinois
Department of Secondary Education

ABSTRACT

During their first class meeting, approximately 200 students enrolled in an introductory course in Pascal for computer science majors were surveyed to determine their prior programming experience and level of initial programming skill. Upon completion of the course, the final course grade distribution was analyzed to establish whether or not students of varying background experience and skill performed equally well during the course. The results indicate that the more experienced and high initial-skill students significantly out performed those students with less experience ($p = .0172$) and skill ($p = .0176$). Additionally, students with prior experience in BASIC were *not* found to be *disadvantaged* relative to students with prior experience in Pascal. Students with prior experience in *both* BASIC and Pascal were found to account for a significantly disproportionate number of the superior grades awarded in the course.

Over the past several years, serious attention has been focused on the issue of the adequate preparation of precollege students for studies in the field of computer science. How should programming be taught? Should it be taught like mathematics, *with* mathematics, like a first or second natural language? Should instructors emphasize grammar and syntax, or composition (problem solving), or both? Is it even desirable that students learn a programming language prior to beginning a formal program in computer science at the college or university level? Should they learn Logo, BASIC, Pascal or some combination of these and/or other languages? As yet, answers for these questions are unclear.

Campbell (1984) found that college undergraduates who completed a preliminary programming course which emphasized problem solving performed significantly better in subsequent introductory level programming courses designed for computer science majors. Dupras, LeMay and Mili (1984) reported that an early emphasis on programming methodology facilitated the learning of a variant of the Pascal language for computer science undergraduates. Based on the results of these studies, early exposures to programming concepts which emphasize problem solving would appear to be beneficial for students. If an early exposure to programming is beneficial, then which language should be the vehicle of that experience? Does it matter?

Although technically, all languages should be considered in a discussion of this type, this paper will focus on BASIC and Pascal, for they are currently at the heart of a controversy among professional computer educators. Should BASIC or Pascal be taught to precollege students? This is a question about which many opinions have been expressed. According to Braun (1983), many computer scientists are so opposed to the BASIC language that they would rather that students *not* be exposed to programming experiences in the BASIC language. This position appears to be widely held among computer scientists (Mallozzi, 1985; Sedlmeyer & Parman, 1980; Ziegler, 1984), but never so strongly as verbalized by Dijkstra (1972):

> It is practically impossible to teach good programming to students that have had a prior exposure to BASIC; as potential programmers they are mentally mutilated beyond hope of regeneration (p. 130).

Others, including Alfred Bork (1985), noted physicist, computer scientist and computer educator, have repeatedly advised teachers not to teach BASIC.

On the other hand, Braun (1983) claims that "students learning to program in BASIC can be taught good programming technique" (p. 30). This position is also espoused by Luehrmann (1983) who states that "The powerful ideas of computing transcend details of language" (p. 12). And, Blaisdell and Burroughs (1985) state that "It is as possible to teach about abstractions, decomposition, and modularization with BASIC as it is with Pascal" (p. 6).

The major points to consider in this computer language controversy can be summarized as follows: (1) style; (2) expense; (3) ease-of-use; (4) proliferation; and (5) teacher preparation. Pascal is a modern programming language which emphasizes modularity and top-down structure. BASIC requires only a linear structural organization which allows poorly planned, and awkward flow of execution. BASIC is encoded in the read-only memory of nearly every microcomputer sold in America today, so it's cost is included in the purchase price of the microcomputer. The Pascal language represents an additional cost to the consumer. Most experts would agree that BASIC is easier to learn to use than Pascal, and since every microcomputer "speaks" BASIC, it is the language most commonly available to independent learners. Since BASIC is older than Pascal, many more of our current precollege teachers who have formal computing experience probably obtained it through a course in BASIC. Also, among those precollege teachers who are self-taught (and they are many), BASIC was the only language widely available on the microcomputer for many years and thus was learned "by default".

A careful examination of these facts reveals why many computer educators support teaching the BASIC language. Although BASIC has significant limitations. it also has compelling practical advantages. Is BASIC as harmful an influence as some suggest? While many computer educators might gladly substitute the Pascal language for the BASIC language on all microcomputers, this is not possible. Thus, for the next several years, BASIC will continue to be learned by millions of precollege learners all across America *whether we teach it or not*! Why? Because it's there!

Will the widespread availability of BASIC adversely affect our future programmers and computer scientists? Are these students doomed to be "mentally mutilated beyond hope of regeneration"? Perhaps, but perhaps not. However, computer educators must recognize that BASIC will continue to be with us for quite some time yet to come. Simply ceasing to teach BASIC in a formal way to the precollege audience (and their teachers!) as some suggest will not make it go away. On the contrary, it will allow even more self-taught practitioners of "street BASIC" to move through the public school system with *no opportunity* to learn sound, structured programming principles in, what may be for some, the language in which they initially conceptualize problem solutions

The purpose of this paper is to present research data gathered in an attempt to answer a pivotal question in this controversy. Is BASIC harmful to the health of potential computer science students? If BASIC programming experiences are "mentally mutilating" the precollege student population, then instruction in the BASIC language should be abandoned in favor of a more suitable programming language. If it cannot be demonstrated that programming experiences in BASIC are harmful to students' future learning efforts, then less emphasis should be placed on abandoning BASIC and more emphasis should be placed on making the best of the current situation by *encouraging* the teaching of structured programming techniques using the BASIC language (Huston, 1985; Luehrmann, 1984).

Methodology

Students taking an introductory course in Pascal for computer science majors (CS 121) at the University of Illinois at Urbana-Champaign during the Spring semester of 1986 constituted the sample population for this study. During the first class meeting, a survey instrument (see Appendix A) was administered to the class to collect background information concerning students' prior programming experiences and programming skill. A total of 208 out of approximately 250 students completed the survey instrument.

As a measure of programming experience, students were asked to list any programming languages with which they had had prior experience. Based on their responses, five types of prior experience were identified--none, BASIC only, Pascal only, BASIC and Pascal, and other (Fortran and Cobol). Students were also asked to demonstrate *minimal* programming skill by using the language in which they had had previous experience to write a short program which would request three numbers, average the three numbers and report the result of the calculation to the user. This simple measure of minimal programming skill was examined as an indication of programming ability. Essentially, this requirement forced the students to "speak" extemporaneously in the computer language of their choice. These student programs were evaluated and categorized as being either correct, incorrect with minor errors, or incorrect with major conceptual errors.

Upon completion of the Spring semester, students' final course grades were obtained and matched wtih the survey data. The final course grades were assigned on the basis of performance on seven machine problem exercises (20%), two one-hour examinations (40%) and the final examination (40%). Examinations assessed content knowledge, problem solving and structured programming techniques. Final course grades were calculated as percentages and these percentages were then catagorized using a seven point grading scale (A+ = 7, A = 6, A- = 5 . . . E = 1).

Inspection of the final course grade percentages revealed that with respect to subgroupings based on initial programming skill and prior programming experience, the data were not normally distributed and would not pass the assumption of homogeneity of variance for analysis by Analysis of Variance techniques. Thus, the catagorical final grade data (A - E) were analyzed using a nonparametric procedure, the Chi-Square test. The Chi-Square procedure was employed to establish whether or not statistically equivalent final course grade distributions existed among the different prior programming experience and initial programming skill subgroups. Significance levels for all analyses were set a priori as $\alpha = .05$.

Results

Analyses of the data collected during this research have yielded the following results. A majority of the students (76%) who enrolled in CS 121 possessed some prior programming experience in the BASIC language (see Figure 1). Among students with no prior experience or "other" experience in programming, course attrition rates were highest--72% and 63%, respectively (see Figure 2). Initial programming skill was found to correlate significantly with final course grade distribution ($N = 100$; $r = .34$; $p = .001$) using the Spearman rank order correlation procedure. Students grouped according to prior programming experience were found to differ significantly ($p = .0172$) on final course grade distributions (see Table 1, Figure 3). Students grouped according to initial programming skill were found to differ significantly ($p = .0176$) on final course grade distributions (see Table 2, Figure 4). Reliability estimates (KR_{21}) for the measures of student achievement used during the semester ranged from .92 to .99. Correlations among the various measures of student achievement used during the semester ranged from moderate ($r = .50$) to strong ($r = .81$) indicating that student achievement was relatively consistent throughout the semester.

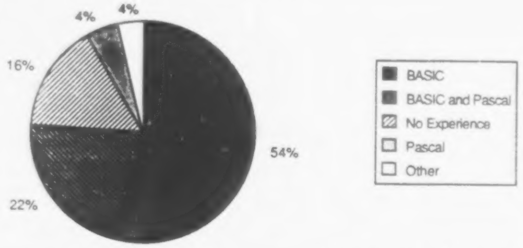


Figure 1. Prior programming experience reported by students enrolled in CS 121.

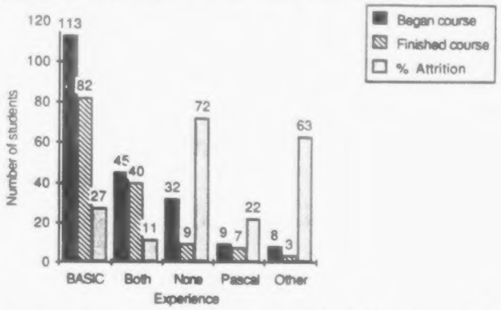


Figure 2. Attrition in CS 121 course (31% overall) broken down by prior experience groupings.

Table 1

Final Course Grade Distributions for Students of Varying Programming Background

Prior Programming Experience in:	Final course grade distribution A+	A	A-	B	C	D	E	AB	Subtotal
BASIC	(8. 1) 4	(16.9) 15	(6.4) 8	(15.7) 16	(21.5) 25	(4.7) 4	(2.3) 3	(6.4) 7	82
BASIC and Pascal	(4.0) 10	(8.2) 10	(3.1) 2	(7.7) 7	(10.5) 7	(2.3) 3	(1.1) 1	(3.1) 0	40
No Experience	(.9) 0	(1.9) 3	(.7) 0	(1.7) 1	(2.4) 3	(.5) 0	(.3) 0	(.7) 2	9
Pascal	(.7) 0	(1.4) 0	(.6) 1	(1.3) 3	(1.8) 2	(.4) 1	(.2) 0	(.6) 0	7
Other	(.3) 0	(.6) 1	(.2) 0	(.6) 0	(.8) 0	(.2) 0	(.1) 0	(.2) 2	3
Column subtotal	14	29	11	27	37	8	4	11	141 Total

Note. Expected frequencies in parentheses
$\chi^2 = 46.06$; d.f. = 28; $p = .0172$

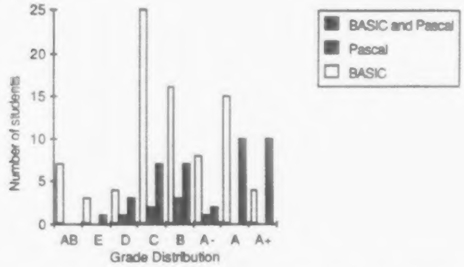


Figure 3. Final course grade distribution broken down by prior programming experience groupings.

Table 2

Final Course Grade Distributions for Students of Varying Initial Programming Skill

Results on Programming Task:	Final course grade distribution A+	A	A-	B	C	D	E	AB	Subtotal
Correct Solution	(7.2) 11	(14.4) 16	(6.0) 6	(12.6) 13	(15) 8	(2.4) 3	(2.4) 1	(3.0) 5	63
Minor errors in solution	(2.1) 1	(4.1) 6	(1.7) 0	(3.6) 3	(4.3) 6	(.7) 1	(.7) 1	(.9) 0	18
Major errors in solution	(2.7) 0	(5.5) 2	(2.3) 4	(4.8) 5	(5.7) 11	(.9) 0	(.9) 2	(1.1) 0	24
Column subtotal	12	24	10	21	25	4	4	5	105 Total

Note. Expected frequencies in parentheses
$\chi^2 = 27.30$; d.f. = 14; $p = .0176$

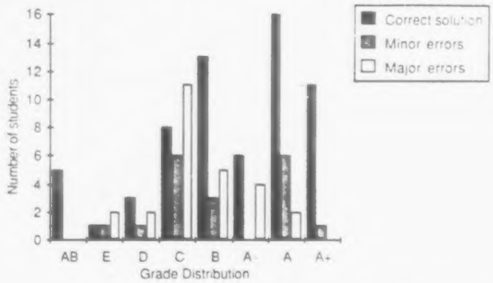


Figure 4. Final course grade distribution broken down by initial programming skill groupings.

Conclusions

The results of this investigation do not support the notion that a knowledge of BASIC--often referred to as "the junk food of computer programming" (Bork, 1985, p. 29)--is "mentally mutilating" (Dijkstra, 1972) or in any other way harmful to potential computer science students. Furthe these data do not support the position taken by some computer scientists (Sedlmeyer & Parman, 1980) that it would be preferable for students to enter introductory computer science courses with no background experience rather than prior experiences in BASIC.

Prior experience, broadly-based and not necessarily language specific, appears to be a highly significant factor associated with student performance in a "typical" introductory course in computer science. This position is supported by the fact that the attrition rate was highest (72%) among those who reported no prior experience, while it was lowest (11%) among those who reported prior experience in both BASIC and Pascal. Also, those students who possessed some prior experience with both BASIC and Pascal accounted for a significantly disproportionate share of superior grades in the course. *If* BASIC were to exert any negative influence upon students' subsequent abilities to master structured programming principles, then students with Pascal-only backgrounds should clearly have been "advantaged" when compared to those students with BASIC in their backgrounds. These data clearly *do not* support the existence of any such advantage. In fact, students with experience in more than one language actually appear to be "advantaged" relative to their less experienced peers.

This same interpretation is supported by the initial programming skill data. Presumably, students with high levels of initial programming skill obtained their skill through either more and/or better programming experiences. These data indicate that students who began the course with relatively high levels of programming skill *in any language*, tended to perform better than did their less skilled peers. Thus, experience, whether measured simply as self-reported "exposure" to one or more computer languages or by actual programming skill, appears to be a very significant factor associated with student performance in an introductory course in Pascal programming.

These results indicate that despite what must surely be tremendous variation in the quality of BASIC (or any) language instruction accomplished in the precollege environment (Linn, 1985), students of BASIC are *not disadvantaged* at the outset of their formal studies in computer science at the university level. Apparently, either programming experiences in BASIC are better now than they were when Dijkstra wrote his often quoted memorandum in 1972, or there is nothing intrinsicly wrong with the language. Therefore, educators should feel free to teach BASIC or any computer language which will afford students opportunities to experience programming as a highly interactive problem solving process. Organization and structure should be encouraged during the problem solving process as well as during the development of algorithmic solutions regardless of the specific computer language employed. These experiences will enable students to gain valuable insights concerning the capabilities and limitations associated with computers and on the proper role for computers in the problem solving process.

BIBLIOGRAPHY

Blaisdell, J. H., & Burroughs, A. (1985). How to tell if a programming language is OK: What's wrong with BASIC for teaching business students how to program? *Association for Computing Machinery: SIGCSE Bulletin*, *17*(3), 5-8.

Bork, A. (1985). *Personal Computers for Education*. New York: Harper & Row.

Braun, L. (1983, September). Pascal and the AP exam: Why ETX was wrong . . . and how teachers and students across the country will pay the price. *Electronic Learning*, pp. 30, 32.

Campbell, P. (1984). The effect of a preliminary programming and problem solving course in performance in a traditional programming course for computer science majors. *Association for Computing Machinery: SIGCSE Bulletin*, *16*(1), 56-64.

Dijkstra, E. J. (1972). *Selected Writings on Computing: A Personal Perspective*. New York: Springer-Verlag.

Huston, A. (1985, June). Structuring BASIC. *Byte Magazine*, pp. 243-248.

Linn, M. (1985). The cognitive consequences of programming instruction in classrooms. *Educational Researcher*, *14*, 14-16, 25-29.

Luehrmann, A. (1983, April). Personal communication. *Electronic Learning*, p. 12.

Luehrmann, A. (1984, May). Structured programming in BASIC, Part 1: Top-down BASIC. *Creative Computing*, pp. 152-156.

Luehrmann, A., & Peckham, H. (1986). *Computer Literacy: A Hands-On Approach*. New York: McGraw-Hill.

Mallozzi, J. S. (1985). A course in programming languages for educational computing. *Association for Computing Machinery: SIGCSE Bulletin*, 17(2), 29, 31.

Sedlmeyer, R. L., & Parman, W. (1980). A college preparatory course in computer programming. *Association for Computing Machinery: SIGCSE Bulletin*, 12(1), 166-171.

Zeigler, W. (1984). The new advanced placement computer science course: An analysis. Association for Computing Machinery: SIGCSE Bulletin, 16(2), 34-36, 47.

APPENDIX A

Computer Experience Survey

You are not required to participate in this survey, but your cooperation will be greatly appreciated. The purpose of this survey is to determine whether or not prior programming experience has any influence on your performance in CS 121. We will need to know your SS# and your final course grade in order to examine this relationship. Your signature below will authorize us to examine your course grade for this specific purpose only. Your instructors will not have access to these survey forms.

Directions: If you have had prior programming experience, please respond to each of the four items below before you leave class today. If you have had no prior programming experience, please respond only to item number one.

1. Signature: ______________________________

   Social Security Number: ______________________

2. Please list any programming languages with which you have had prior experience.

COMPLETE EITHER 3a OR 3b.

3a. If you have had prior experience with BASIC and/or Pascal, then choose either BASIC or Pascal, depending upon the one you know, or the one with which you feel most comfortable, and write a program to complete the exercise described below.

3b. If you have only had prior experience with languages other than BASIC or Pascal, then choose the language with which you feel most comfortable, and write a program to complete the exercise described below.

-------------------------Program Exercise----------------------------

On the back of this page, write a short program which, when executed, will:

1. request that three numbers be input,
2. calculate the average of the three numbers, and
3. report to the "user" the calculated average.

---

4. Which of the three responses listed below indicates your agreement with the following statement:

   "I am confident that the program I just wrote will execute without error."

   1. Disagreee
   2. Undecided
   3. Agree

Contact: Michael L. Waugh
University of Illinois
Dept. of Secondary Education
Champaign, Illinois 61820

NUTRITION AND METABOLISM FOR NURSES:

BASIC CONCEPTS IN NUTRITION

George Kalnitsky
Department of Biochemistry
University of Iowa
Iowa City, IA 52242

ABSTRACT

This program attempts to acquaint the reader with the scientific basis of nutrition. The basic components in an adequate diet are discussed, followed by the metabolism of glucose, fat and proteins, their conversion to energy or to fat, and the roles of vitamins, minerals and enzymes. After discussing obesity and the potentially serious problems involved in using "magical" weight loss methods, we present ideas for sensible weight loss.

INTRODUCTION

People suffering from nutritional problems are found in all parts of the world, in affluent, technologically-advanced countries, as well as in poor, developing countries. There are three basic nutritional problems which are responsible for this suffering: (1) undernourishment, (2) malnourishment, and (3) overnourishment. In conditions of undernourishment, people simply do not have enough to eat. This is a serious problem in that an estimated ten to twenty million people starve to death each year throughout the world, and approximately 460 million people (about 1 out of 6) are undernourished. Malnourished individuals lack nutrients that are important for the proper functioniong of the body. The lack of vital nutrients at critical times in the development process can do irreversible damage. Early malnutrition which stunts growth has been shown to impair subsequent memory, ability, and behavior in experimental animals. To the extent that brain growth is impaired concurrently with early retardation in linear growth, the population at risk includes more than one half of the world's children. In the problem of overnourishment, it is possible to get all of the food you need, and all of the right kinds of nutrients, but to eat too much. This has been linked to a variety of serious health problems, including arteriosclerosis and high blood pressure.

These nutritional problems have a variety of political, economic, sociological, and even psychological causes. This program is primarily concerned with one important contributing factor--ignorance. Nurses, as the largest single group of health professionals, are in an ideal position to intervene in these problems. Therefore, this program was designed to improve nurses' knowledge base about nutrition and metabolism, so that they will be better able to distinguish, and help others distinguish, between dietary recommendations that are frivolous, or even harmful, and those that have a firm basis in nutritional science.

This computer-assisted program is also designed for students and professionals in the health occupations and for lay persons who are interested in food and nutrition and who are trying to assess the merits of all the dietary information we are being constantly deluged with in books, magazines, pamphlets, handouts, newspapers, radio and television. The course consists of a manual and one diskette.

OVERVIEW

Objectives

This program is designed to enable the reader to describe and identify important and basic concepts of nutrition. The concepts include:

*why we eat
*the basic nutrients in our food
*how we digest and utilize these food nutients
*the essential compounds in our diets, and why they are essential
*what constitutes proper nutrition
*some problems with obesity and "magical" weight-loss methods
*how to approach a sensible weight-loss diet

Program Description

This program introduces the user to the basic nutrients in our diet, such as carbohydrates, fats, proteins, vitamins, minerals, water, and fiber.

The program defines and identifies these basic nutrients and goes on to show how the body utilizes them in normal daily situations. It illustrates what each of these nutrients do; how they work; why they are important; their relationship to each other; and how enzymes, vitamins and minerals function in the breakdown and synthesis of sugars, fats, and proteins.

This knowledge of which nutrients are essential and why they are essential leads to an understanding of what makes a good diet and why it is a good diet. It also leads to an intelligent and knowledgeable approach to the problems of obesity and fasting. After presenting examples of several "magical" weight loss methods, fad diets, and diet pills (indicating why they don't work and the possible dangers involved), the program describes some guidelines for proper nutrition and safe and effective weight loss.

Students and health professionals should be able to utilize this program or information in this program for their classroom and other methods of teaching. For lay persons who are interested in food and nutrition and in keeping themselves healthy, this program should help them to assess the merits of all the dietary information with which they are constantly being deluged. It will give them a factual basis to enable them to judge what they hear and read, to be a better consumer, and to correct misconceptions and mis-information on the road to better health.

THE COMPUTER PROGRAM

The Menu and General Outline for this computer-assisted instruction program is as follows:

A. Introduction
   Importance of nutrition.
   Nutritional problems (undernutrition; malnutrition; overnutrition). Why do we need to eat.

B. The basic components in an adequate diet.
   The bulk nutritients: carbohydrates, fats and proteins.
      Chemistry, energy values and functions of each; the role of the liver in maintenance of blood glucose; essential fatty acids and essential amino acids.
   Vitamins, minerals (bulk and trace elements) and water.
      The role of dietary fiber.

C. How the body uses food.
   The metabolism of glucose, fat and protein; conversions to energy and fat. The roles of vitamins, minerals and enzymes. Feasting and fasting.

D. Diet and balance.
   Obesity. Serious problems involved in "magical" weight-loss methods.
      Fad diets (low carbohydrate, high protein, high protein-high fat). Quick weight loss diets. Diet pills (starch blockers, amphetamines.
   Sensible weight loss.
      Energy budgets. Food groups. How can you determine that your diet is indeed proper for you?

In addition, the Menu contains a HELP? option that allows novice users to obtain on-screen directions for using the program. Users are given further control through the use of sub-menus. For example, the second section, "The Basic Nutrients" contains eight sections: Introduction, Carbohydrates, Fats, Proteins, Vitamins, Minerals, Water, and Fiber. Each section in the program ends with a summary.

The program menu includes:

A. Introduction
B. The Basic Nutrients
   1. Introduction
   2. Carbohydrates
   3. Fats
   4. Proteins
   5. Vitamins
   6. Minerals
   7. Water
   8. Fiber
C. How the Body Uses Food
   1. Introduction

    2. Glucose Metabolism
    3. Lipid Metabolism
    4. Protein Metabolism
    5. Vitamins, Minerals and Enzymes
    6. Feasting and Fasting
- D. Diet and Balance
    1. Obesity
    2. Magical Weight Loss Methods
        a. Fad diets
        b. Starch blockers
        c. Other pills
    3. Safe and Effective Weight Loss
    4. Proper Nutrition

The Development Process

As an inexperienced author of computer-based instructional materials, the first step was review of several existing programs (in other subjects) to become familiar with the capabilities of computers and to plan how the computer could be used best with the chosen topic. Next, came the outline of content to identify essential material and appropriate sequences. Once the outline was finished, work with an instructional designer and the programmers began. In many cases, the lack of background in this subject on their part was an advantage. Truly central concepts were more readily identified, and examples necessary for their understanding were incorporated into the program. Other factors which were developed collaboratively included screen design, use of color, motion and graphics, and provisions for user interaction. At this point, the editorial associate for the project entered the text and programming notes into our PRIME word processing system and produced hardcopies for review by the authors, programmer, instructional designer and an evaluator. The text was edited based on these reviews, and the actual coding, using a pseudo-BASIC language began. After coding, a software package called PRE translated the pseudo-code into the dialect of BASIC for the IBM-PC, where the machine-specific routines such as the graphics were added.

Graphics and animation were used in this program to clarify basic physiological and biochemical principles. For example, in illustrating how the liver regulates blood sugar levels within narrow limits in normal conditions, after a meal, or during a period of fasting, diagrams with motion were used to show the rise and fall of liver glycogen levels concurrently with changes in blood sugar levels. In between meals, normal levels of glycogen and blood sugar are illustrated. After a carbohydrate-rich meal, the blood glucose level rises but is quickly returned to normal as the blood glucose is enzymatically converted to glycogen in the liver and stored for future use. If a person were to skip lunch and keep working, the body tissues remove glucose from the blood, causing the blood glucose level to decrease. However, this blood glucose level would not go down very far because it would be replaced by glucose obtained from liver glycogen. The result is that the content of liver glycogen decreases and the content of blood glucose is returned to normal and maintained within relatively narrow limits, at normal levels.

User interaction was maintained by dispersing questions throughout each section. The final section of the program (in the manual) is a self-test over the entire program.

RECOMMENDED READINGS

The author used the following books (and other journal articles) as sources for his software program. For more extensive reading in the area of nutrition, biochemistry or dieting, please consult these textbooks and journals:

Bender, A.E. & Bender, D.A. (1982). Nutrition for Medical Students. New York, : John Wiley and Sons.

Devlin, J.M. (ed.) (1986). Textbook of Biochemistry with Clinical Correlations (2nd ed.). New York: John Wiley and Sons.

el Lozy, M. (1980). Nutrition. Kalamazoo, MI: The Upjohn Co.

Kritchevsky, D. et. al. (1985) Symposium: Physiological and Metabolic Effects of Dietary Fiber. Proc. Soc. Exp. Biol. Med. 180, 407-452.

Lehniger, A.L. (1982). Principles of Biochemistry. New York: Worth Publishers, Inc.

McGilvery, R.W. (1983). Biochemistry: A Functional Approach (3rd Ed.). Philadelphia: W.B. Saunders Co.

Pennington, J.A.T. & Church, H.N. (1980). Bowes and Church's Food Values of Portions Commonly Used (13th ed.). New York: Harper and Row.

Montgomery, R., Dryer, R.L., Conway, T.W. and Spector, A.A. (1983). Biochemistry: A Case-Oriented Approach (4th ed.) St. Louis: C.V. Mosby.

Smith, E.L., Hill, R.L., Lehman, I.R., et al (1983). Principles of Biochemistry (7th ed.), Vol. I General Aspects; Vol. 2, Mammalian Biochemistry. New York: McGraw-Hill, Inc.

United States Department of Agriculture (1975). Handbook of the Nutritional Contents of Foods. New York: Dover Publications.

Whitney, E.N. & Hamilton, E.M.N. (1984). Understanding Nutrition (3rd Ed.). St. Paul, MN: West Publishing Co.

ADCIS Proceedings, November 1986

IS IT WORTH IT? DETERMINING THE COST-EFFECTIVENESS OF USING COMPUTER-ASSISTED INSTRUCTION IN NURSING EDUCATION

Donna E. Larson

Associate Professor of Nursing
Grand Valley State College
Allendale, Michigan 49401

ABSTRACT:

A method for analyzing the costs and benefits related to using computer-assisted instruction (CAI) in nursing is presented. Six cost-benefit components are identified and discussed: hardware, software, degree of usage and lifespan, installation and maintenance, effectiveness and efficiency of instruction, and nature of students and instruction. Finally, an example is presented to illustrate how costs-benefits for a nursing CAI system can be ascertained. This paper was originally an invited presentation at the International Symposium on Nursing Use of Computers and Information Science, Calgary, Alberta, Canada, May 1-3, 1985.

INTRODUCTION:

When analyzing the costs and benefits related to using computer-assisted instruction (CAI) in nursing, it is essential to first identify the major cost-benefit components: (1) hardware, (2) software, (3) degree of usage and life-span, (4) installation and maintenance, (5) effectiveness and efficiency of instruction, and (6) nature of students and instruction.

HARDWARE:

When CAI in nursing was first developed twenty years ago, hardware costs were substantial. Early nursing CAI was dependent upon large centralized computer systems which controlled numerous terminals. Moreover, because much of nursing CAI was distributed via large networks (e.g. PLATO), on-going telecommunication charges added greatly to the costs of implementing CAI. Early nursing CAI required both a substantial initial investment and a continuing financial commitment for network communication. In the 1960s and 70s, a large, mainframe, time-sharing computer system cost an average of $10,000 (U.S. dollars) per terminal, plus approximately $800 per month for communication charges (Braun, 1980).

Recent technological advances in computers, particularly the advent of microcomputers, have considerably reduced the hardware cost component of educational computing. A typical microcomputer system currently costs between $1000 and $5000, depending on the system's capabilities and peripheral devices. Because of the stand-alone usage pattern of microcomputers, on-going telecommunication costs are eliminated.

SOFTWARE:

Although the hardware costs are usually the most visible costs of educational computing, the software costs are the largest overall expense. Commercially available nursing software continues to be relatively expensive, although the prices of some nursing software marketed by major U.S. publishers have decreased by as much as 50% in 1984. As competition among the software publishers increases, improvements in quality and decreases in cost are projected to continue. Current purchase prices range from approximately $100 for individual nursing CAI programs to approximately $3500 for nursing authoring systems.

Even though commercially available nursing software is relatively expensive, individual institutional development of software is even more costly. The actual costs of developing software are difficult to estimate, as the costs depend on the complexity of the content and instructional design, the skill of the author, and so on. However, rough estimates of the average costs of software development in higher education are between $2750 to $3000 per hour of student instruction (Van Der Drift, Langerak, Moonen, & Vos, 1982). The high costs are primarily attributable to the large number of person-hours of development time required. Kearsley (1982) reports that from 100 to 400 hours of development time are required for each hour of student instruction produced. Larson (1981) reports that it required 120 person-hours for her development of a nursing CAI simulation which requires on the average of ten minutes of student learning time. In a recent survey conducted by Hodge (1983), the person-hour requirements for developing CAI materials for higher education ranged from 10 to 714 hours per hour of student interaction. Hodge (1983) reports that the person-hour CAI development time requirements are decreased by 50% when a lesson is developed based upon previously developed lessons. It is anticipated that the use of authoring systems will dramatically decrease the person-hour time required for developing nursing CAI.

DEGREE OF USAGE AND LIFESPAN:

Computer-assisted instruction differs from traditional classroom-based instruction in that CAI usually involves higher capital costs (for hardware and courseware development) but relatively low on-going or operating costs. Consequently, the extent to which use of the system can be spread out over students and time is a critical aspect of instructional computing cost analysis (Kearsley, 1982). The number of students who use a computer system is a function of its availability. It is necessary to carefully project student use in order to calculate the hardware cost per student hour of instruction. Since nursing education requires that students be "off-campus" for clinical learning for some proportion of their educational experiences, the availability of instruction is a particularly important variable to consider when attempting to compute the cost of CAI in nursing education. Kearsley (1982) cautions that one cannot assume that the computer system will be fully utilized at all available times. He reports that most instructional computer systems have student utilization rates of about 50%. Therefore, the computers are used about half of the time that they are available.

Another important factor related to cost is the anticipated lifespan of the hardware. Because of the rapid development in computer technology, most hardware has an effective lifespan of three to five years before it is considered obsolete (Kearsley, 1982; Spuck, 1981). Therefore, it is important to amortize the costs of hardware over the anticipated lifespan in order to project realistic yearly costs.

The anticipated lifespan of software should also be considered. The nature of nursing software determines the anticipated lifespan. For example, if the content area is critical care nursing, it can be anticipated that the lifespan of that particular software is shorter than software in the content area of basic skill development, decision-making, or process application. As with hardware, it is important to amortize the costs of software over the anticipated lifespan in order to realistically project yearly usage costs.

INSTALLATION AND MAINTENANCE:

In contrast to large, mainframe computer systems, installation costs for microcomputers are usually minimal. Some additional expense may be necessary if additional electrical outlets and sufficient wiring do not already exist. Some furniture (such as work tables at correct height) and other minor accessories (such as chalkless boards, dust covers, and anti-static floor mats) may also need to be purchased.

The use of large-scale integrated circuit technology has increased the reliability of microcomputers. Maintenance contracts with microcomputer vendors typically annually cost 7-15% of the original purchase price. As Pressman and Rosenbloom (1984) report, the maintenance of peripherals (such as printers, plotters, disk drives, monitors) is frequently more problematic than maintenance of the microcomputers. Such peripherals are often subject to heavy student use and abuse. Maintenance contracts on peripherals typically cost 15-25% of the original purchase price annually.

EFFECTIVENESS AND EFFICIENCY OF INSTRUCTION:

A number of studies have been

conducted which compare the effectiveness and efficiency of CAI with conventional teaching methodologies in nursing (Bitzer, Boudreaux, & Avner, 1973; Collart, 1973; Donabedian, 1979; Droste-Bielak, 1980; Huckabay, Anderson, Holm, & Lee, 1979; Kirchhoff & Holzemer, 1979; Larson, 1981; Schleutermann, Holzemer, & Farrand, 1983). These studies found that nursing students using CAI were able to learn the subject matter equally as well as their peers using more traditional instructional modes. Additionally, in most of the studies cited above, students were able to learn the material in 1/3 to 1/2 the amount of time with CAI when compared to the conventional instructional strategies. The findings of the nursing research are consistent with studies in other disciplines. In a meta-analysis of 54 research studies conducted on the use of CAI in higher education, Kulik, Kulik, and Cohen (1980) concluded that CAI was as effective but more efficient in terms of student learning-time than conventional instuctional methodologies. Savings in student learning time can also be extrapolated to savings in instructor teaching time.

Spuck (1981) points out that compression of student learning time has important economic significance, both from the standpoint of institutional costs as well as student expense. If a baccalaureate degree program could be compressed from four years into three, then the student would require one less year of institutional support (social subsidy), student tuition and fees, and lost income. Spuck maintains, however, that the credit-hour generation would remain constant, so student-generated income to the institution would remain the same.

An extensive literature search revealed only one study which investigated not only the comparative efficiency and effectiveness of nursing CAI, but also the cost. Larson (1981) found that a CAI simulation was as effective as conventional skills laboratory "live" instruction for teaching calculation and regulation of intravenous flow rates, a skill with both cognitive and psychomotor components. She found that the cost for the CAI was $1.44 per student, while the cost for the skills laboratory instruction was $2.01 per student. The difference in cost was mainly attributable to decreased faculty teaching time and use of expendable laboratory materials.

NATURE OF STUDENTS AND INSTRUCTION:

Different types of students cost different average amounts to educate. For example, Kearsley (1982) states that the cost per student hour for educating elementary school students is typically less than $1, while the cost of educating college undergraduates ranges from $2 to $5 per student hour. However, the cost of educating professional students (such as medicine, nursing, law, and engineering students) is often in excess of $10 per student hour. Because of the demonstrated savings in student learning time and instructor teaching time, CAI has the potential for substantially reducing the overall costs of professional nursing education.

Some aspects of nursing education are particularly well-suited for CAI. For example, those parts of nursing education which are highly repetitive and instructor-intensive in nature (such as skills lab instruction), "foundation" level content/process which is so important for continued growth (such as priniples of nursing process application), very important basic level concepts which must be mastered prior to clinical application (such as dosage calculations, medical terminology, and so on), clinical learning opportunities that are rare for student exposure or potentially dangerous for patients (such as cardio-pulmonary resuscitation, or the application of therapeutic communication techniques to a suicidal patient), and higher-level clinical decision-making skill development (such as working with very complex, multi-problem patients and families, leadership and management decisions, and so on) are all areas which are "ripe" for CAI.

TOTAL NURSING CAI SYSTEM COSTS: AN EXAMPLE

The above discussion of the major cost-benefit components of CAI in nursing provides the background for discussing an example of total nursing CAI system costs. The example projects the costs of setting up and operating a simple microcomputer laboratory and implementing CAI for a medium size nursing program (total of 150 students in the clinical portion of the major).

The costs are broken down into CAI components and amortized over a four-year period. These yearly costs are then divided by the number of hours of projected student use, to obtain an estimate of average CAI costs per student hour. Projected student use takes into account the off-campus, clinical nature of nursing education, wherein students are not likely have access to CAI for two days out of every five. In the example it is assummed that each student will spend an average of 1 1/2 hours per week using CAI (See Table I.).

Table 1. Four-year Projected Nursing CAI Costs for an Inexpensive System

Number	Description	Unit Cost	x 4 Years if Variable Cost	4 Years Total Cost
10	Apple IIe microcomputers Apple 13" color monitors Double disk drives	$1800	---	$18000
2	Epson FX-80 printers	450	---	900
	Optional accessories	400	---	400
		Total Hardware Costs:		$19300
	Software Costs (various commercially available nursing software packages)	$5000	x 4	20000
	Installation Costs	2000*	---	2000
	Maintenance Costs (Hardware and Software)	3000	x 4	12000
		Total Variable Costs:		$34000
		TOTAL CAI SYSTEM COSTS:		$53300
	AVERAGE COST PER YEAR (TOTAL DIVIDED BY 4 YEARS):			$13325

Yearly Student Use:

Assume: 150 students x 1/2 hour/day x 3 days/week x 36 weeks = 8100 yearly CAI hours

Then: $13325 (average cost per year) divided by 8100 (yearly CAI hours) = $1.64 average CAI cost per student hour

---

* Installation costs in this example assume no major construction, but do include the purchase of "computer-height" student work surfaces and the addition of simple electrical wiring.

If the microcomputer equipment in the example were upgraded to IBM-Pcs ($3500 per unit) rather than Apples, the average cost per CAI hour would be $2.16. Both the $1.64 and $2.16 estimates of average cost per CAI hour are fairly consistent with other estimates in the literature ($1.32 for Pressman & Rosenbloom, 1984; $1.50 for Sojka, 1981; and $2.88 for Larson, 1981). CAI costs are approximately 20% of the cost for traditional instructor-intensive instruction; the gap should increase in the future as conventional instructional costs rise and CAI costs continue to fall.

Nurse educators must insure that CAI is used where most appropriate in nursing curricula --- where repetition is frequent, where patient risk is high, where skill attainment is basic to further development, where higher level clinical decision-making skills need to be practiced and improved, where learning experiences might not otherwise be available. CAI should not be viewed as the panacea for all of the problems facing nursing education today. Rather, CAI should be used as an effective, efficient, and cost-saving instructional strategy to meet specific nursing curricular needs.

References

Bitzer, M., Boudreaux, M., & Avner, R.A. (1973 February). Computer-based instruction of basic nursing utilizing inquiry approach. Urbana, IL: Computer-based Education Research Laboratory.

Braun, L. (1980 July). Computers in learning environments: An imperative for the 1980's. Byte, 5(7), 7-10.

Collart, M.E. (1973 August). Computer-assisted instruction and the teaching-learning process. Nursing Outlook, 21, 527-532.

Donabedian, D. (1979 April). Computer-taught epidemiology. Nursing Outlook, 24, 749-751.

Droste-Bielak, E. (1980). The use of computer simulation in teaching interviewing techniques to beginning nursing students in a baccalaureate program. Unpublished doctoral dissertation, University of Michigan.

Hodge, G. (1983). Estimation of the person-hour requirements for developing computer-assisted instructional lessons. Unpublished masters thesis, Ohio State University.

Huckabay, L., Anderson, N., Holm, D., & Lee, J. (1979 July/August). Cognitive, affective, and transfer of learning consequences of computer-assisted instruction. Nursing Research, 28, 228-233.

Kearsley, G. (1982). Costs, benefits, & productivity in training systems. Reading, MA: Addison-Wesley.

Kearsley, G. (1984). Training and technology: A handbook for HRD professionals. Reading, MA: Addison-Wesley.

Kirchhoff, K., & Holzemer, W. (1979 March). Student learning and a computer-assisted instruction program. Journal of Nursing Education, 18, 22-30.

Kulik, J., Kulik, C., & Cohen, P. (1980 Winter). Effectiveness of computer-based college teaching: A meta-analysis of findings. Review of Educational Research, 50, 525-544.

Larson, D. (1985). Cost-effectiveness of educational computing in nursing environments. In Hannah, K., Guillemin, E., & Conklin, D. (Eds), Nursing uses of computers and information science (pp. 217-221). Amsterdam: North-Holland.

Larson, D. (1981). The use of computer-assisted instruction to teach calculation and regulation of intravenous flow rates to baccalaureate nursing students. Unpublished doctoral dissertation, Michigan State University.

Pressman, I., & Rosenbloom, B. (1984). CAI: Its cost and its role. Journal of Educational Technology, 12(3), 183-208.

Schleutermann, J.A., Holzemer, W.L., & Farrand, L.L. (1983 October). An evaluation of paper-and-pencil and computer-assisted simulations. Journal of Nursing Education, 22, 315-323.

Sojka, D. (1981 November). CAI catches on. Datamation, 27, 188-190.

Spuck, D.W. (1981 Fall). An analysis of the cost-effectiveness of CAI and factors associated with its successful implementation in higher education. AEDS Journal, 15(1), 10-22.

Van Der Drift, K.D., Langerak, W.F., Moonen, J.M., & Vos, P. (1982). The choice of a CAL system. Computers and Education, 6(4), 361-368.

Computers in Nursing: An Annotated Bibliography, 1979-1984 Using ZyIndex

Barbara Thomas, Ph.D.
Professor, The University of Iowa
Programmer: Steve Bowers, B.A.

Abstract

The purpose of Computers in Nursing: An Annotated Bibliography is to provide nurses and others with easy access to the literature on computers in nursing published from 1979 through 1984. The package uses ZyIndex to search, display and print summaries of articles located by a variety of search strategies. The manual which accompanies this package describes these search options and also gives users assistance in modifying the package for similar tasks or in producing additional annotated bibliographies. The source code for our program is included in the manual.

Introduction

This project was started several years before funding was acquired, but was put aside because external support was needed to complete the project. When the Division of Nursing awarded the three year Special Projects Grant (Grant#D10 NU27075) in July, 1983, a decision was made to cover the five year period, 1979-1984, so that a package could be completed, refined and distributed within the grant period. Yearly updates are planned.

The 477 entries of the bibliography are contained on two floppy diskettes. The files on this diskette have already been indexed with ZyIndex. The Index List Disk is included in this package. Using ZyIndex's search requests, files on any topic regarding computers and nursing can be retrieved to be viewed on the screen or printed.

We checked and re-checked entries to ensure the accuracy of the files, but we readily acknowledge that information beyond these abstracts may be desired by some users. The full citations are supplied. The Manual was designed to provide users with adequate information for using the package. Since this system was based on ZyIndex, repetition of information from the ZyIndex Manual was inevitable. We are grateful to ZyLab for allowing us to do this, and we encourage users to consult the ZyIndex Manual for further information and clarification.

Purpose of the Annotated Bibliography

The primary purpose of this project was to provide nurses and others with ready access to the literature on computing in nursing, and to use microcomputer technology to do it. The items in this bibliography were located using Medlines, CINAHL and ERIC searches.

It is hoped that this bibliography will serve a number of needs. First, individuals can search topics related to their clinical interests to see what software development or evaluation research has been done. Second, nurse administrators charged with developing nursing information systems can gain rapid access to published reports of existing systems and also the processes and policies found most efficient and effective. Third, the production of review articles on various topics will be facilitated. Fourth, nurses who are novices in computing can locate help easily and quickly. Nurse educators can also use the bibliography to see what the prophets in nursing say about the future of computer applications. And lastly, the bibliography will, it is hoped, stimulate an awareness of the broad impact of computer technology on nursing among nurses, nursing students and other health professionals.

Development of the Annotated Bibliography Program

The programmer, Steve Bowers, made a concerted effort to develop a system which can be used for other needs, for example, additional bibliographies. In doing this, he reviewed various database systems and chose ZyIndex as meeting our needs best as well as being flexible. The Annotated Bibliograpy program can be modified in any number of ways, the only requirements being that we be notified of changes made and that such modifications cite the source and be used only locally. The Source Code is included in the manual.

This package is designed for use with an IBM-PC microcomputer (or IBM-XT, IBM-AT or 100% IBM-compatible microcomputers), ZyIndex, and a printer. The printer is not essential if the user merely wishes to read the entries on the screen. The PC must be booted up with DOS 2.0 or higher.

Searches may be initiated using certain codes or words, phrases, wild card symbols, search connectors and/or parentheses. Search requests can be up to 160 characters long (two lines). A code of one or two characters may be searched, but if the codes are not used, each word in the search request must be at least three characters long and no longer than sixteen characters. Simple searches, e.g., those using the codes of the two code system, are very fast. The more complex the search, the longer it takes to retrieve the files. One may search for numbers as well as codes, words, and/or phrases. Searches are interrupted whenever the search words are not found in the index list.

A two code system was developed to devise categories for sorting all possible articles on computers in nursing. The first level refers to the type of article, and the second to the specialty topic of the article. The first set of categories, the types of articles are as follows:

A. Advocacy or position papers - These articles discuss the advantages, disadvantages, usefulness, limitations, and possible new application of computers in nursing.

B. Prescriptive or descriptive narrations - In this category, the articles report specific applications which have been or should be developed. Thus, "how to do it", "how we did it," and "how it should be done" papers are included in this class.

C. Informative articles about computer literacy or computer technology - These articles provide readers with basic information about computer technology. They furnish nurses with facts and explanations designed for a nurse audience. They do not report original work or new applications.

D. Research and evaluation - These articles describe evaluation, evaluation research or research about computer applications. Included are studies about specific applications and also meta-analysis papers. No attempt was made to judge the rigor of the studies.

E. Review article - Descriptions and discussions of secondary sources of information about computers in nursing are included here.

F. Software - The entries describe specific software and the capabilities of each one or the developmental processes used. Entries from the Software Exchange in *Computers in Nursing* were placed here.

The second set of categories is as follows:

1. Nursing administration/the business of providing nursing care - This category includes articles which focus on recordkeeping, financial matters, staffing and patient records.

2. Nursing practice/pediatrics - These articles report clinical applications related to the care of infants and children.

3. Nursing practice/obstetrics - Articles in this category deal with applications concerning childbirth preparation and post-partum care, as well as labor and delivery.

4. Psychiatric nursing - Entries contain references to any phase of psychiatric nursing: assessment, planning care, providing care, and evaluation of outcomes.

5. Nursing practice/medical-surgical - Most articles in this category deal with nursing care provided in acute care facilities and nursing care of clients with chronic illnesses.

6. Community nursing - These articles discuss health promotion and maintenance activities that are provided by nurses in community settings and out-patient clinics.

7. Long-term care - These articles focus on rehabilitation, maintenance and long-term care of special populations, especially geriatric nursing.

8. Nursing education - A wide variety of instructional applications are included in this category. Articles deal with both the teaching and administrative aspects of nursing education, e.g., instruction, record keeping, advisement, scheduling, and testing.

A search of the files without using the codes can be made easily if certain rules are followed. The contents of such a search request must include at least one content word or phrase, and may also contain wild card symbols, search connectors, and parentheses. Let's look at each of these components in turn.

Using a content word is the simplest type of search request. Entering the word <diet> can retrieve all articles containing the words "diet" and "diets". A content phrase contains two or more words without any connectors between them. For example, if you are investigating the topic, "discharge planning", the search request <discharge planning> retrieves only those articles with the word "discharge" followed by the word "planning".

Wild card symbols allow for variations of the content words so that all forms of the topic word will be retrieved. There are two wild card symbols: <?> and <*>. The first one, <?>, is a single letter wild card symbol and the second, a multi-character wild card symbol is used following a root. When a root word is followed by <*>, ZyIndex will expand its search to include all content words that begin with the root.

Search connectors serve to specify more than one content word and identify the relationship between or among them. The four ZyIndex connectors are: OR, AND, NOT, and W/n. "OR" enlarges the search topic, and is useful in retrieving articles which contain words related to each other but not in root form. For example, obstetric OR labor OR delivery will retrieve all of the articles containing any of these words.

"AND" narrows the search topic. Such a search request as <neonatal AND intensive AND care> instructs the computer to retrieve only those articles which contain all three of the words. "AND NOT" also narrows the search topic. For example, for a search concerned with administration but not patient administration, the search request, <administration AND NOT patient> should be entered. "W/n" helps define abstract concepts where n is a number in the "within" connector. It helps the user find files which contain specified words contained in certain types of phrases. For example, <computer W/10 printer> retrieves all articles containing the words computer and printer within ten words of each other.

Parentheses allow more complex searches to be initiated by grouping several content words and connectors. For example, the entry, <(cardi* OR heart) AND monitor> retrieves any file which contains the words, heart, cardiac, cardiology, cardiologist, cardiovascular, etc. plus monitor.

The equipment and software requirements are as follows:

1. DOS 2.0 or higher - to boot the PC.
2. Index List Disk - this diskette contains the location of every content word and number in the database.
3. ZyIndex Program Disk - this diskette contains the ZyIndex Search program and the program to "index" files into ZyIndex.
4. Two diskettes containing files indexed into ZyIndex: (1) 79Group 1 to 84Group 2. This diskette contains files NBIB.1 through NBIB.291; (2) 84Group 2 through 84Group 3. This diskette contains files NBIB.292 through NBIB.477. The diskette contain spaces for the addition of more files.
5. Microcomputer system with two disk drives -- ZyIndex runs on the IBM-PC, PC-XT, PC-AT computers, and all 100% IBM-compatible computers. A minimum of 256K of RAM memory is

Procedures for using the package will be demonstrated.

ADCIS Proceedings, November 1986

A RESEARCH-BASED STATISTICAL MATRIX FOR DEMONSTRATING IMPACT OF COMPUTER-BASED LEARNING

Dr. James L. Morrison

University of Delaware

ABSTRACT

Using data collected during an instructional experiment utilizing computer-based modules reflecting content in an introductory consumer economics course at a university, a research-based statistical matrix is demonstrated for analyzing findings based upon preliminary statistical analysis. The methodology described provides researchers with a tool for illustrating similarities/differences among sample populations. Variations between the performance of students in control and experimental groups may be identified by plotting data collected onto a two dimensional matrix, resulting in a procedure for demonstrating study impact.

INTRODUCTION

Over the past decade, a series of computer-based modules have been planned, designed, and developed for use on the CYBER mainframe as part of the University of Delaware timesharing network. The series is labeled, "Consumer in the Marketplace," and is delivered as a segment of an introductory consumer economics course offered to sophomores each semester. An instructional experiment was conducted to determine the impact these lessons have upon student achievement in consumer economics.

PROCEDURES OF EXPERIMENT

The hypothesis tested during the experiment was there is no difference between the test scores of those students using computer-based lessons and those who do not. In testing this hypothesis, 63 students enrolled in an introductory course in consumer economics at the University of Delaware were divided into two groups; a control group whose members received the traditional instructional approach and an experimental group whose members completed computer-based lessons in addition to routine instructional activities.

The computer-based lessons were grouped into three categories reflecting the primary segments of the course. These groupings are:

Introductory Modules:

OVERVIEW OF CONSUMER DECISIONING PROCESS

1. Consumption
2. Optimal Information
3. Sovereignty
4. Time-Probability
5. Consumer Rights
6. Rationality

Focus Modules I:

INCREASING CONSUMERS' EARNING POWER

7. Investment-in-Human-Capital
8. Opportunity Costs
9. Indexing

Focus Modules II:

INCREASING CONSUMERS' SPENDING POWER

10. Transfer of Income
11. Product Benefits/Tradeoffs
12. Consumer Systems
13. Product Liability
14. Optimal Consumption Stream

The labels assigned to the modules reflect concepts that are presented and analyzed during each lesson. Contemporary issues are also included for providing examples of how a particular concept may impact upon consumer behavior in the marketplace.

Students in both the control and experimental groups received *identical* instruction which was delivered by the same instructor, read the same texts, and were provided identical supplementary reading assignments. In addition, each group of students were tested on the same day using the same testing instrument for each of the three parts of the course coinciding with the three groupings of computer-based modules indicated in the above list.

A Pearson Coefficient of Correlation was computed to determine the degree of relationship between a student's overall grade point average and the three numerical scores for each of the three tests administered. In addition, the Chi-Square (x^2) statistical measure was used to determine if computer-based modules were a significant factor in any differences in student achievement in the control and experimental groups. The significance level tested was the .10 level of confidence.

USING A STATISTICAL MATRIX TO DOCUMENT FINDINGS

Traditionally, researchers rely upon a variety of statistical measures for documenting findings and deriving conclusions. In this experiment examining the impact of computer-based modules upon student achievement, such a tradition is followed. However, in addition, a statistical matrix has been utilized to clarify findings. A discussion of each is provided below:

Statistical Documentation. Identical tests were administered to both the control and experimental groups. The data was then placed into a 2 by 3 table for statistical measurement purposes. The results of the statistical procedure are provided below:

(1) *Impact of Computer Modules Upon Student Achievement in Part I of Course*:

In Table 1, a x^2 of 5.21785715 was significant at the .10 level of confidence indicating that the hypothesis be rejected, supporting the position that the computer modules are a significant factor in differences in tests scores of the control and experimental groups.

(2) *Impact of Computer Modules Upon Student Achievement in Part II of Course*:

In Table 2, a x^2 of 5.99535281 was significant at the .05 level of confidence indicating that the hypothesis be rejected, providing statistical evidence that computer modules are a significant factor in differences in student achievement between the control and experimetal groups.

(3) *Impact of Computer Modules Upon Student Achievement in Part III of Course*:

In Table 3, a x^2 of 6.0449296 was significant at the .05 level of confidence indicating again that the hypothesis be rejected, illustrating that computer modules are playing a role in differences in test scores between the control and experimental groups.

Utilizing a Statistical Matrix for Demonstrating Impact. While the satistical measures used have disproved the hypothesis tested as part of the experiment, it remains important to analyze "where" the differences in test scores are occurring. For example, insight as to whether computer lessons are impacting more favorably on one group versus another (e.g., students with high intellectual capacity versus low intellectual capacity) or whether the difference is spread systematically among all intellectual capacities of students. In order to gain such insight, a two-dimensional matrix may assist by depicting where differences are occurring and where computer lessons are having the most impact.

For example, Graph #1, reflecting data of the *experimental* group using computer lessons, presents data matching a student's overall grade point average (e.g., 2.0 to 3.7 on the y axis) and individual test scores (e.g., 0 to 100 on the x axis). By plotting individual scores on a matrix, a representation of what is occurring emerges. In Part I, of course, the evidence indicates that students with overall grade point averages of 2.5 or lower were generally helped more when using computer modules than those with averages 2.6 or above. Graph #2, reflecting data of the control group which did *not* have access to computer lessons, illustrates the finding that students who did not use the computer lessons systematically had lower test scores no matter what their grade point average indicated.

Moreover, when comparing test scores of students in the experimental (Graph #3) and the control (Graph #4) group in Part II of the introductory course in consumer economics, the evidence indicates that test scores in the experimental group again are systematically higher than those in the control group. However, in addition, the evidence also appears to indicate that students with lower grade point averages (e.g., 2.5 or below) in the experimental group had scores closer in proximity to those above 2.5; whereas, in the control group, there was a greater disparity in test scores based upon a student's grade point average.

The conclusion to be drawn here is that students with higher past achievement are apparently doing well whether the computer lessons are used or not. However, the computer lessons appear to assist students with lower grade point averages (2.5 or below) more than those with higher grade point averages. A similar finding was demonstrated on a matrix for Part III of the course in consumer economics. (To avoid reiterating this point, data related to Part III of course will not be included.)

CONCLUDING REMARKS

The use of a statistical matrix for reporting findings based upon prior statistical measurements (in this case, the x^2) gives the researcher insight as to where differences occur. With the ability of today's microcomputers to plot such data quickly and easily, this technique has great potential for analyzing data. In this experiment, those students who have experienced less success at the University appear to be helped the most when computer lessons are utilized in conjunction with an introductory course in consumer economics. Since variations between the performance of students in the control and experimental groups were quite different in Part I and Part II as illustrated in this document, the insight derived by plotting data collected on a two dimensional matrix provides additional evidence important to the study impact.

TABLE 1

COMPARISON OF TEST SCORES FOR PLATO AND

NON-PLATO GROUPS FOR PART 1 OF COURSEWARE

TEST SCORES	PLATO	NON-PLATO	TOTALS
80 - 100	13	2	15
70 - 79	19	9	28
0 - 69	10	10	20
TOTALS	42	21	63

Chi - Square: 5.21785715

Degrees of Freedom: 2

TABLE 2

COMPARISON OF TEST SCORES FOR PLATO AND NON-PLATO GROUPS FOR PART 2 OF COURSEWARE

TEST SCORES	PLATO	NON-PLATO	TOTALS
80 - 100	28	5	33
70 - 79	12	7	19
0 - 69	5	5	10
TOTALS	45	17	62

Chi - Square: 5.99535281

Degrees of Freedom: 2

TABLE 3

COMPARISON OF TEST SCORES FOR PLATO AND NON-PLATO GROUPS FOR PART 3 OF COURSEWARE

TEST SCORES	PLATO	NON-PLATO	TOTALS
80 - 100	34	9	43
70 - 79	5	5	10
0 - 69	3	4	7
TOTALS	42	18	60

Chi - Square: 6.0449296

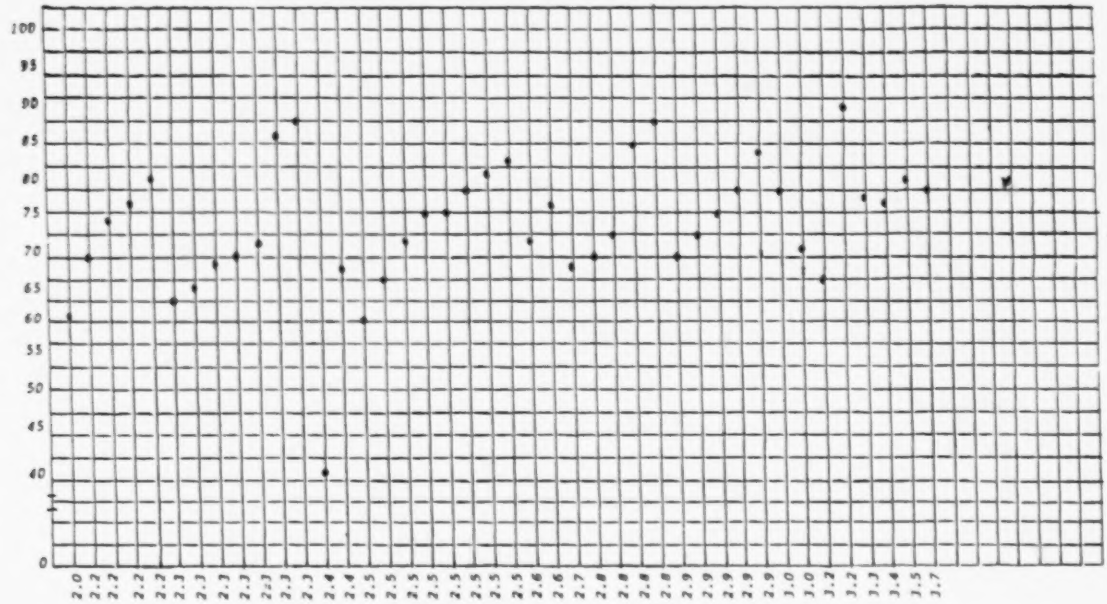
Degrees of Freedom: 2

GRAPH 1

SCORES FOR TEST 1 FOR PLATO GROUP

BY STUDENT OVERALL CUMULATIVE AVERAGE

N=42



Mean: 74.0

Standard Deviation: 8.88956642

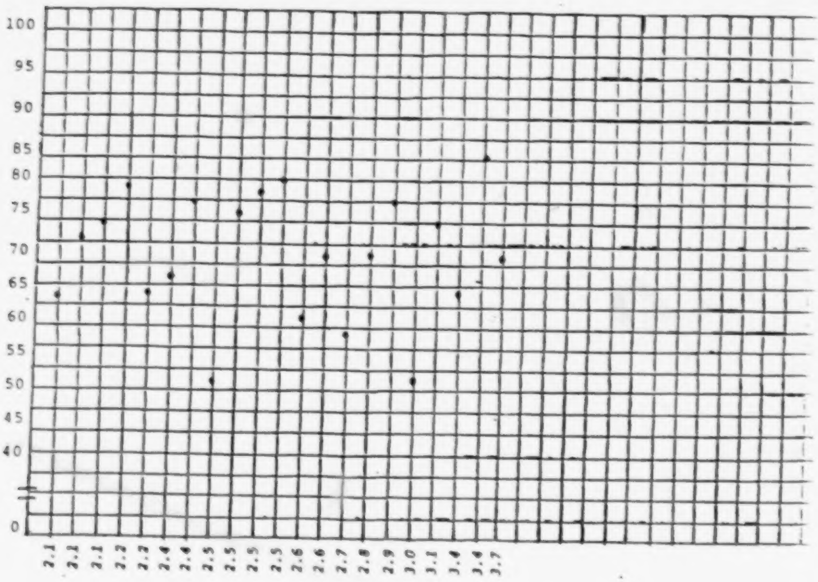
Pearson Coefficient: .300179144

GRAPH 2

SCORES FOR TEST 1 FOR NON-PLATO GROUP

BY STUDENT OVERALL CUMULATIVE AVERAGE

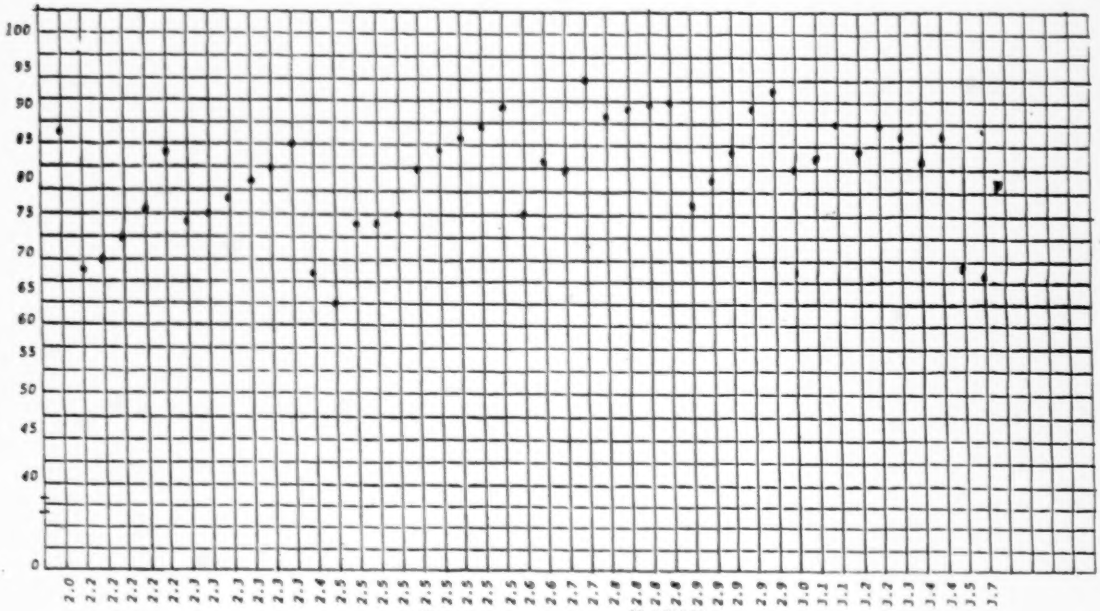
N=21



Mean:	69.4761904
Standard Deviation:	8.93095164
Pearson Coefficient:	.0597689879

GRAPH 3

SCORES FOR TEST 2 FOR PLATO GROUP
BY STUDENT OVERALL CUMULATIVE AVERAGE
N=45



Mean: 80.177778

Standard Deviation: 7.44766252

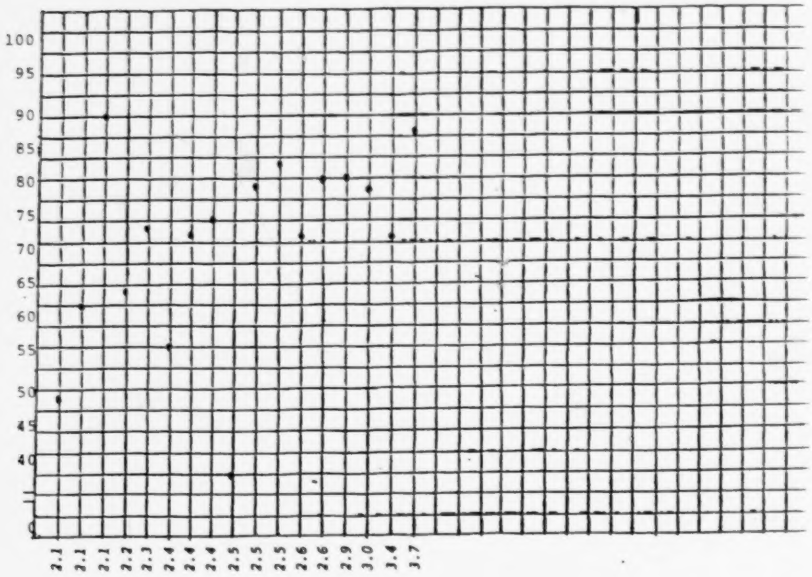
Pearson Coefficient: .185201507

A RESEARCH-BASED IMPACT OF COMPUTER-BASED LEARNING

GRAPH 4

SCORES FOR TEST 2 FOR NON-PLATO GROUP
BY STUDENT OVERALL CUMULATIVE AVERAGE

N= 17



Mean:	71.1764706
Standard Deviation:	13.9294441
Pearson Coefficient:	.335147915

Contact: James L. Morrison
College of Human Resources
University of Delaware
323 Alison Hall
Newark, DE 19716

302-451-8541

ADCIS Proceedings, November 1986

USING THE COMPUTER TO ASSIST IN THE MANAGEMENT OF A CBT PROJECT

Adele S. Suchinsky and David R. Losleben

Department of Defense

ABSTRACT

This paper describes the use of commercial applications packages to assist in the management and design of a Computer Based Training project. These packages were integrated with project management techniques to track the flow of the instructional design process. By using a computer to generate project documentation, screen design pages, and scheduling tables, team members and tasks were followed with flexibility and ease.

INTRODUCTION

The background of this project is similar to projects found in government agencies, industries, and schools. Our requirement for developing a CAI course was one of necessity and cost savings. A large population of students in diverse locations needed a common training program. All courseware had to be presented to the trainees on an IBM-PC with a monochrome monitor.

The organization of the project was the responsibility of a Project Manager, who had the constant task of monitoring the progress of the project and reallocating resources when necessary. His duties included scheduling completion dates, tracking progress of team members, coordinating team review sessions, monitoring development of course documentation, and evaluating module designs. In order to complete the project within the scheduled time frame, two instructional designers, working concurrently, were used. Additional team members consisted of two coders, a subject matter expert, and a graphics person. Coordination was needed to monitor a project that involved a development team of six members and over three hundred tasks. To achieve maximum flexibility in managing a project of this size, several commercial software packages were used.

Three areas were identified where a computer could be utilized to assist in the management and design of the project: Documentation, Project Planning, and Project Management.

DOCUMENTATION

The structure of the project was defined in preliminary meetings attended by the Project Manager, instructional designer, and the subject matter expert, who determined the breakdown of modules, lessons within these modules, and the behavioral objectives and associated test items. Paper documentation was needed for a course outline, objectives and test items, necessary forms and the evaluation plan.

This documentation was produced using a word-processing package that incorporated features that facilitated not only hard-copy formatting, but also text editing. The package used was pfs:Write because it was readily available, but any word processing program would do. Additional packages used to refine documents included pfs:Proof as a spelling checker, and Nice Print which enhanced the dot matrix print into Near Letter Quality copy.

To create the flowcharts used in a CAI project, a program such as Interactive Easyflow may be used. By inputting information in an interactive manner, the selection of flowchart symbols and the relationship of one element to another can be formed into a printed or plotted chart showing the desired lesson direction. The ease of creating these charts and the related time savings over the laborious hand drawn charts makes the computer a valuable resource, especially during the initial design phase where most of these decisions need to be made.

PROJECT PLANNING

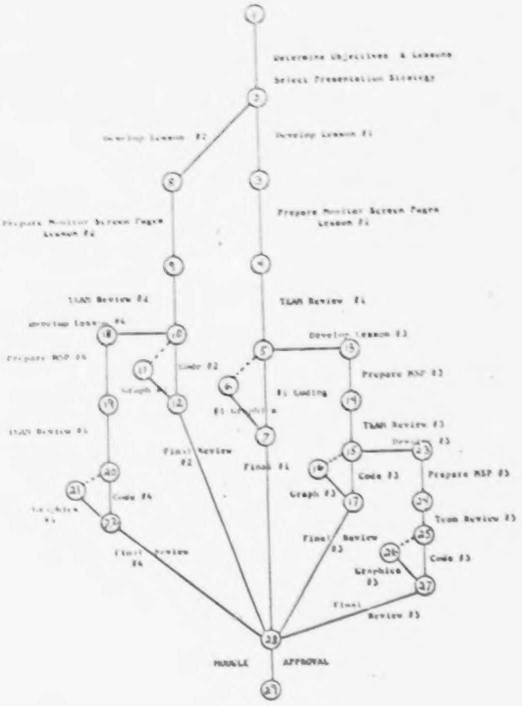
The initial phase of the CBT project was to determine a design strategy to insure the flow of ideas and tasks. Figure 1 shows the particular strategy used in the project. Projecting the completion time for one lesson to be fifteen days, the development stages were plotted in relation to the time alloted for each. For example, the development of monitor screen pages for one lesson would take six days, beginning on the second day with completion by the eighth day (this became the building block for the task development worksheet which would follow).

```
 Development
   Stages
---------------------------------------------------------------
1  Objective,XXX
   Test Item,XXX  (1 Day)
   Verify,   XXX

2  Monitor      XXXXXXXXXXXXXXXXXXXXX
    Screen      XXXXXXXXXXXXXXXXXXXXX  (7 Days)
     Pages      XXXXXXXXXXXXXXXXXXXXX

3  Team                              XXX
    Review                           XXX  (1 Day)

4  Coding                               XXXXXXXXXXXX
                               (4 Days) XXXXXXXXXXXX

5  Graphics                             XXXXXX
                                        XXXXXX (2 Days)
6  Final
   Review/                                          XXXXXX
   Revision                              (2 Days)   XXXXXX
---------------------------------------------------------------
Day ->          1  2  3  4  5  6  7  8  9  10 11 12 13 14 15
```

Fig. 1

With a firm design strategy in mind and development tasks known, the next step was to correlate these stages with known tasks. Figure 2 represents a pictorial view of the development of a five lesson module as it would appear on a task worksheet. The numbers in the circles are reference locations which are used to identify particular tasks. For example, the reference location 3-4 represents the task for preparing monitor screen pages for Lesson 1. Note the concurrent development of Lesson 2 at reference location 8-9. No software package was identified to create this chart and it was done manually.

With tasks identified and graphed, data collected in figures 1 and 2 was transferred to a spreadsheet format so that work assignments and time allocations for each team member could be calculated. Scheduling conflicts were noted and simply resolved by adjusting the time frame of one team member according to the needs of the whole team (Figure 3).



Determine Objectives & Lessons
Select Presentation Strategy
Develop Lesson #2
Develop Lesson #1
Prepare Monitor Screen Pages
Lesson #2
Prepare Monitor Screen Pages
Lesson #1
TEAM Review #2
TEAM Review #1
Develop Lesson #4
Develop Lesson #3
Prepare MSP #4
Code #2
Graph #2
#1 Coding
#1 Graphics
Prepare MSP #3
TEAM Review #4
TEAM Review #3
Final Review #2
Final #1
Develop #5
Code #3
Prepare MSP #5
Code #4
Graph #3
Graphics #4
Team Review #5
Final Review #3
Code #5
Final Review #4
Graphics #3
Final Review #5
MODULE
APPROVAL

Figure 2

```
Day ->     1     2     3     4     5     6     7     8     9    10     11     1?

Work :
Asgn :
 V   :
GRPH :                                                          4--5   6------?
     :                                                          9-10
CDR1 :                                                          9-10   10---->1?
     :
CDR2 :                                                          4--5   5------>7
     :
SME  : 1--2  2--3                                               4--5   10-13
     :       2--8                                                      10-15
ID#1 : 1-2   2--3  3-----------------------------------------4 4--5   10-13  13-->
     :
ID#2 : 1-2   2--8  8-----------------------------------------9 9-10   10-18  13-->
     :
PM   : 1-2                                                      4--5
     :                                                          9-10
      ------------------------------------------------------------------------------
Day ->     1     2     3     4     5     6     7     8     9    10     11     12
```

Figure 3

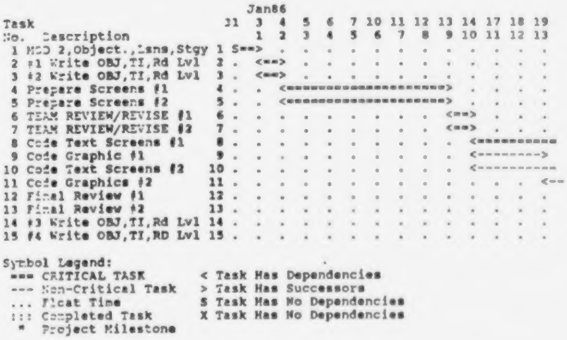
A spreadsheet program, such as Lotus 1-2-3, allowed for adjustments in the positions of tasks, in relation to the team needs for particular members during revision/evaluation sessions. Tasks, responsibilities, and time allocations now clarified, the data was ready to be put into project management format.

PROJECT MANAGEMENT

Nearly all of the elements needed to follow the development of the project were now gathered. Tasks and team members were identified, the design strategy was established, and task assignments were made. The next, and final stage was to put this information together into a workable format.

The remaining information to be collected was the projected work calendar. This was where the computer turned possible weeks worth of work into hours. By using a project management program called Pro-Ject 6, this data was manipulated and calculated. The beginning date was entered and a five day work week selected. All holidays, vacation periods, and known prior committment dates were also entered so an available working calendar would be as accurate as possible. Having entered team members and costs per day along with the tasks into the computer, this information could now be processed to match the projected calendar with the projected work and available resources.

A chart was printed for paper copy (Figure 4). With the use of a utility package called Sideways, the chart was displayed lengthwise, with the projected time frames and due dates for all team members to see. It was flexible enough to print either large or small characters.



```
                                     Jan86
Task                              31  3  4  5  6  7 10 11 12 13 14 17 18 19
No.  Description                      1  2  3  4  5  6  7  8  9 10 11 12 13
 1 MOD 2,Object.,Lsns,Stgy  1 S==>  .  .  .  .  .  .  .  .  .  .  .  .
 2 #1 Write OBJ,TI,Rd Lvl   2  .  <==>  .  .  .  .  .  .  .  .  .  .  .
 3 #2 Write OBJ,TI,Rd Lvl   3  .  <==>  .  .  .  .  .  .  .  .  .  .  .
 4 Prepare Screens #1       4  .  .  <======================>  .  .  .  .
 5 Prepare Screens #2       5  .  .  <======================>  .  .  .  .
 6 TEAM REVIEW/REVISE #1    6  .  .  .  .  .  .  .  .  .  <==>  .  .  .
 7 TEAM REVIEW/REVISE #2    7  .  .  .  .  .  .  .  .  .  <==>  .  .  .
 8 Code Text Screens #1     8  .  .  .  .  .  .  .  .  .  .  <==========
 9 Code Graphic #1          9  .  .  .  .  .  .  .  .  .  .  <-------->
10 Code Text Screens #2    10  .  .  .  .  .  .  .  .  .  .  <----------
11 Code Graphics #2        11  .  .  .  .  .  .  .  .  .  .  .  .  .  <--
12 Final Review #1         12  .  .  .  .  .  .  .  .  .  .  .  .  .  .
13 Final Review #2         13  .  .  .  .  .  .  .  .  .  .  .  .  .  .
14 #3 Write OBJ,TI,Rd Lvl  14  .  .  .  .  .  .  .  .  .  .  .  .  .  .
15 #4 Write OBJ,TI,RD Lvl  15  .  .  .  .  .  .  .  .  .  .  .  .  .  .

Symbol Legend:
 === CRITICAL TASK        < Task Has Dependencies
 --- Non-Critical Task    > Task Has Successors
 ... Float Time           S Task Has No Dependencies
 ::: Completed Task       X Task Has No Dependencies
  *  Project Milestone
```

Figure 4

By viewing the chart essential questions could now be answered: Could the project be done in the allocated time? Were enough people assigned to the project? Were enough resources available? Was more planning time necessary? The computerized project manager had projected a project into the future with all the known problems entered into its files. The final decisions and tracking problems, however, still remained with the Project Manager. Now there was a visual referent available to aid in these decisions and it was capable of being updated quickly and easily.

SUMMARY

Management of a CBT project can be enhanced with the introduction of computers and specialized software. Consider the massive editing job needed when one word is misspelled...throughout the project. Or an index, where every occurrence of a specific item is needed. Or the ability to project a time deadline, or calculate project costs, within hours rather than days. Each task is monumental in itself, but with computer support each task becomes minimal. By taking simple, readily available packages, you can simplify the organization and management of tasks and resources with flexibility and ease. It is not necessary to become "locked in" to any particular package as those described here. Other packages are available which will be able to fill your needs.

Above all, the task of CBT project management depends upon a lot of up-front planning, realistic deadlines, and correct software and hardware utilization.

SOURCES

The software packages listed below are those used in the project just described. There are many commercial packages on the market which may be used to accomplish project management goals. Similar resources may be identified in popular trade journals and magazines.

1. pfs:WRITE, Software Publishing, Inc., Mountain View, CA

2. pfs:PROOF, Software Publishing, Inc., Mountain View, CA

3. Interactive Easyflow, HavenTree Software Limited, Thousand Island Park, NY

4. LOTUS 1-2-3, Lotus Development Corp., Cambridge, MA

5. PRO-JECT 6, SoftCorp, Inc., Clearwater, FL

6. Sideways, Funk Software, Cambridge, MA

7. NicePrint, Spies Laboratory, Gardenia, CA

ADCIS Proceedings, November 1986

Institute for Computer Assisted Learning:

An Alternative Approach to Implementing CAL in a University

W. Bruce Clark
Institute for Computer Assisted Learning
The University of Calgary
Calgary, Alberta, Canada

Abstract

There are any number of universities who can be regarded as leaders in the implementation of computer assisted learning on their campuses. This paper describes an approach to the implementation of CAL by a relative late-comer to the technology. It would be unfair to suggest that the University of Calgary has done nothing in the past relative to CAL. In fact the Computer Applications Unit of the Faculty of Education was an early adopter of the technology; and there have been many instances of individual faculty members who have developed and used CAL in various forms as part of their teaching. Yet the institution as a whole has only begun to address the potential of CAL for the delivery of instruction. Because it is coming to this relatively late, much of the romance associated with earlier implementation has vanished and a number of concerns are having to be addressed--what systems are appropriate, what effect will time spent on developing CAL materials have on promotion and tenure, what will be the long-term effect of substantive moves in the direction of CAL on university resources? A partial response to these issues was the creation of the Institute for Computer Assisted Learning, and this paper will describe the circumstances surrounding the creation and operation of this Institute for the first 18 months of its existence.

BACKGROUND

The University of Calgary is located in Calgary, Alberta, in the foothills of the Canadian Rockies. It is one of four provincial universities and has 16 faculties and about 16,000 students in undergraduate and graduate programs. It has extensive instructional computing resources including a CDC Cyber 205 supercomputer, a Cyber 830 which runs PLATO, a Honeywell multics system, and several Digital VAXs. The campus computers are networked through Develnet, and are accessible from terminal pools in several locations on campus, from many offices, and through dialup. For all its computing resources, only minimal use has been made of them for either computer managed instruction or for computer assisted instruction. The PLATO system is a recent installation which will support approximately 80 simultaneous users after sufficient communication ports are installed. In looking at microcomputers, the University has adopted the position that it will support three brands of microcomputer: Zenith, IBM, and Macintosh. It is further moving in the direction of making it possible for microcomputer pools on campus to also access any of the main frame computers including the PLATO system.

Within this environment two events took place which eventually brought about the creation of an Institute to address computer assisted learning. The first was a special committee appointed by the General Faculties Council (GFC) to study CAL and bring in recommendations concerning its possible implementation on campus. The second was the decision of the University of Calgary to participate in a partnership with other educational institutions in the Calgary area along with elements of the private sector to develop a centre of excellence in computer assisted learning. The Canadian Centre for Learning Systems came into being in January, 1985, and included as the founding partners the University of Calgary, Mount Royal College, the Southern Alberta Institute of Technology, both the Public and Catholic school boards, together with Control Data, Honeywell Information Systems, and Reid Chartwell Ltd. It was necessary for the University to identify some agency to be the link between it and CCLS--a university institute appeared to be an appropriate vehicle to accomplish this linkage.

At the same time the committee of General Faculties Council arrived at the conclusion that an institute would be the best vehicle to achieve what it perceived needed to be done. In studying the role of CAL on the campus it arrived at certain conclusions: (1) there would be resistance among the faculty to the introduction of any "top-down empire" on campus; (2) there was a need to address directly the concerns that legitimate effort spent developing CAL materials or revising course structures to take advantage of CAL would be recognized as meritorious by promotion and tenure committees; (3) there was still a need to address computer literacy issues for faculty in addition to assisting with CAL; (4) there would be resistance to the creation of any agency that would be perceived to be drawing resources away from existing departments at a time when the fortunes of the oil-rich west were starting to turn and the University was being forced into a period of restraint.

The desire of the University administration to have a linking agent to CCLS and the desire of the GFC committee to have an agency to address the CAL needs on campus were merged, and in January, 1985, the Institute for Computer Assisted Learning (ICAL) was created with a mandate to do both. For a number of reasons, not least of which was shortage of space on campus, ICAL was housed off-campus in the Canadian Centre for Learning Systems. It was given a small sum of money for start-up funds, and the author was appointed half-time director.

Getting Started

Given a half-time director, limited funds, some space, and a mandate that encompassed the universe, where does one begin? The first three months were spent trying to get a handle on the problem--this was compounded by the fact that CCLS was starting up simultaneously and at times it was very difficult to sort out which problems belonged to whom. The major issues which appeared necessary to address first were (1) deciding upon where ICAL would focus its strengths, (2) building up some credibility, (3) solving the money problem, and (4) establishing some useful presence on campus. In looking at the potential needs on campus, it was decided that ICAL would make instructional design its particular focus. Given that the development of CAL products require subject matter expertise, instructional design, and programming and that subject matter expertise would come from elsewhere and programming services could be purchased from other units on campus, instructional design seemed appropriate. At the end of three months the staff was extended beyond a half-time director to include an administrative secretary and the first instructional designer.

With the focus established, credibility and money were the next issues to be addressed. Here the association with CCLS played an important part. CCLS was established under the Centres of Excellence Program of the Federal Secretary of State, and as such opened doors to some research and development projects. Furthermore, the location off campus contributed to allowing the Institute to develop, and gain experience and confidence without being under the direct scrutiny of University of Calgary Faculty. The fact that it was beginning to generate its own revenue meant that it was not draining resources away from other University departments and even gained a degree of sympathy by virtue of proving that it could indeed support itself. In the first year of operation four projects were completed; two research and two development projects. Three of these were done with CCLS and included a project funded by the National Research Council of Canada to conduct a feasibility study of using NATAL for authoring CAL in a wide-area environment, plus instructional design support for some business education software and for an ESL (English as a second language) program to assist in preparation for the TOEFL test. A research project developed and piloted a computer augmented approach to cognitive education with Native students.

Coming into its second year of existence ICAL had sufficient project work to cover all its operating expenses. It is participating in a research project developing a knowledge based system for designing computer based training materials, and has a number of projects including one to develop computer assisted instruction to introduce remote users to the University of Calgary's online catalogs. It has also added a second instructional designer to its staff along with a computer programmer. As an aside on the latter appointment, it is interesting to note that the original decision to purchase programming services proved less than satisfactory. From the point of view of project planning and control it has proved more efficient to have a programmer on staff rather than being at the mercy of another agency. Also the model for courseware development has changed. Originally it was perceived that a subject matter expert (and instructional designer) would develop the specifications for a CAL product and hand them to the programmer for coding. This has changed to a perception that a programmer should be involved in the development of the specifications, and this, of course, is most easily facilitated by a unit having its own programmer.

Fellowship Program

Thus far, this paper has described the evolution of ICAL as a unit which supports itself through R & D projects. The first year and a half of its operation has not been totally devoid of on-campus activity. The best known part of that activity is the sponsorship of a fellowship competition for faculty members of the University of Calgary. Fellowship programs are a common component of the programs of institutes at the University of Calgary. The fellowship program at ICAL is intended to achieve two goals; (1) to provide support for individual faculty members who wish to engage in research, development or exploration in the area of CAL, and (2) to partially address the issue of such activities being treated as meritorious by promotion and tenure committees. For the first two years the competition has offered two categories of fellowships which have been designated for lack of a better description "Category A" and "Category B". Neither fellowship provides direct payment of funds to the fellows; rather funds are made available to the fellow's department to purchase release time from a portion of the teaching assignment.

Category A fellowships are intended for candidates with some experience in CAL who might use the tenure of the fellowship to engage in research on some aspect of CAL or develop a worthwhile CAL product for use within the University. This category provides funding for up to two-thirds release time from teaching duties for two consecutive semesters. There is also an agreement with the departments involved that the fellows will be relieved of administrative duties during the tenure of the fellowship.

Category B fellowships are intended for faculty members who have no experience with CAL to spend up to two-thirds of one semester at the Institute examining the potential applications of CAL to teaching their particular disciplines. In their applications, Category B fellows are required to propose a program of activities in which they wish to engage. ICAL itself has developed its own set of activities for these fellows which can be adjusted to suit the fellows' specific needs. That baseline program consists of

- Series of readings
- Hands-on experience:
  - Computer Literacy
  - PLATO
  - Exploration of alternative CAL systems
  - Design workshop
  - Learning one authoring tool
- Trip to a university that is well-established using CAL in the discipline
- Production of a short CAL product
- Report to faculty/department

On the whole the fellowship program must be judged a success. There are things that have been learned from the the two competitions so far that are worth noting and will certainly inform some modifications to the program in the future. For example, it has not proved satisfactory to have research proposals competing with development proposals in Category A. From the point of view of the Institute the less expensive Category B fellowships probably have the greater impact upon the campus in that the results of the inquiry are also disseminated to the fellows' department. This is consistent with the philosophy of ICAL that the most appropriate way to address computer assisted learning within the Faculty of the University of Calgary is at the grass-roots level.

In terms of funding, the University has begun an endowment fund to support CAL on the campus and interest off the endowment is being used to support the fellowship program.

"Moving Back" to Campus

While the Institute for Computer Assisted Learning has been growing in credibility as a Research & Development unit, the primary purpose for its creation is still to foster the use of computer assisted learning in instruction on campus. Since its inception ICAL has provided casual consulting to faculty and departments, but beginning in the spring of 1986 it began to make more substantive moves to address that part of its mandate. For example, it is in the process of developing a series of workshops for faculty members on computer assisted learning. The first workshop offered was at an intermediate level and focused upon instructional development for computer assisted learning. Two others being developed are one that will provide a general introduction to computer assisted learning and another in advanced instructional design of CAL. It has been asked to develop a special workshop for members of the library staff and also to develop a word processing workshop for faculty. The latter is somewhat outside the realm of computer assisted learning, but is consistent with a request from General Faculties Council that ICAL also address some general issues of computer literacy. It should also be noted that it is ICAL's intention that, once workshops are developed, evaluated and revised to a satisfactory degree they will be passed on to others at the University or at CCLS to conduct in future iterations. This way the Institute is freed to move on to other needs or requests.

It is also encouraging in this second year of operation to find the Institute being invited by other parts of the University to provide assistance, and to find that there is a willingness to cover the costs associated with its participation. For example, the proposal to develop a computer assisted learning program to teach remote users about the University of Calgary's online catalogs originated with the Library. ICAL has also been invited to participate in planning for a substantive expansion of the use of computer assisted learning in the Faculty of Humanities. The University of Calgary has recently received a grant to operate a disability information system for Canada, and ICAL has been asked to develop the training materials for that system. The increasing willingness by the University to defray some of the operating costs associated with the Institute frees additional resources from revenue-generating activities for direct assistance to the campus.

In Summary

It appears at this point that the approach taken by the University of Calgary in creating the Institute for Computer Assisted Learning has merit--certainly for this particular institution. It is permitting the "grass-roots" development of CAL to go on without being a disruptive presence; it is in fact supporting the "grass-roots" approach. It is proceeding without being a threat to other units on campus either in siphoning off resources that could be used by other departments or creating a competitive organization. Finally it is beginning to serve also as an agent that links those on campus involved in CAL. It produces an (irregular) bulletin passing on news of who is doing what, what new equipment is coming on the market, what resources are available, and other such information. And it has brought together interested people on campus to plan a conference on CAL in Higher Education to be held in May of 1987.

All this does not suggest that the Institute's approach is without problems. When a unit is expected to be self-supporting, only a limited portion of its resources are available to address its primary mandate. This counter-balances the freedom that comes from not being dependent upon institutional support. When a small low-cost unit is started, there is a time lag while it gathers experience and develops credibility. Finally, when the approach is made to support the growth of CAL from the bottom up rather than from the top down there is a lack of visibility which lends itself to the perception that not much is being done. The Institute for Computer Assisted Learning has certainly experienced all of these problems. It has, however, received considerable support from the administration of the University, and has its own Policy Advisory Board to guide its development and activities.

It is probably too soon to make any pronouncements about the long term success or lack thereof of ICAL's approach to the implementation of CAL at the University of Calgary. Its description here is offered as one possibility among many to promote the appropriate use of this technology in a university. In the University of Calgary, at least, it appears to be working.

Contact: W. Bruce Clark
Institute for Computer Assisted Learning
The University of Calgary
2500 University Drive N.W.
Calgary, Alberta T2N 1N4
Canada

403-270-3303

ADCIS Proceedings, November 1986

INTERFACE BETWEEN AUTHORING SYSTEMS AND PROGRAMMING LANGUAGES

- USER-DEFINED COMMAND -

Yasuhiro Morimoto, Hiroshi Kato, Hiroyasu Chimura,
Masahiko Kurata and Takahiro Sato

C&C Information Technology Research Laboratories, NEC Corporation

Abstract

This paper describes the concept and usage of "user-defined commands" as a user friendly interface between authoring systems and programming languages. This concept is incorporated into the LESSONWRITER authoring system on a microcomputer. The users are able to make BASIC language subroutines as commands on the authoring system. With this function, the parts which necessitate programming languages can be made into modules and can be made transparent. User-defined commands are applied effectively to simulation and to a wide range of sophisticated presentations.

1. Introduction

Many CAI authoring systems have been developed to facilitate the CAI lesson development and revision (Kearsley, 1982). They enable authors to produce CAI lessons by themselves, without knowledge of computer programming. Authoring systems are also a great help to authors who have programming skill, especially for content creation. It is troublesome and awkward, even for sophisticated programmers, to create and revise instructional displays using programming languages.

It is pointed out, however, that authoring systems have functionality limitations (Bork, 1985; Kearsley, 1982). The following criticisms are often made: Presentation patterns are fixed and not flexible, limited mostly to the tutorial lessons; Animation or sophisticated graphics cannot be presented; Simulation is very difficult to implement. Some of these problems can be solved, to some extent, within the range of authoring systems. However, lessons requiring calculation and variable use, such as simulations and complicated animation, necessitate the help of programming languages.

This does not mean that such lessons should be created only with a programming language. The calculation necessity may be only a small part of the lessons. A combined use of an authoring system and a programming language is recommended. This paper describes a user friendly interface between authoring systems and programming languages: user-defined command.

2. User-Defined Command Concept

Some authoring systems allow authors to use programming languages; in most cases, products of those authoring systems are integrated into the programming languages, or they are chained to another independent computer program (Fairweather et al., 1984). This paper suggests, on the other hand, that authoring systems should be main, and that programming languages should be subordinate and integrated into the authoring systems. The parts that have to be written in a programming languages should be made into modules and should be able to be used as commands in the authoring system. Thus, the authors can develop the lessons without being conscious that a programming language has been used, though the process of programming may be done by programmers. This idea is similar to assembly language subroutine calls in high level programming languages. The different point is that the numeric world of programming languages has to be connected to the graphic world of authoring systems. To realize this interconnection, parameters connecting the two worlds should be graphically specified. This concept is realized as "user-defined commands" on the LESSONWRITER authoring system.

3. LESSONWRITER Authoring System

Before describing the user-defined command function more specifically, the LESSONWRITER authoring system (Chimura et al., 1985; Kurata et al., 1986; Morimoto et al., 1986) is briefly mentioned.

LESSONWRITER is a stand-alone format CAI system on a 16-bit NEC microcomputer PC-9801. The LESSONWRITER authoring system is made up of the frame creation subsystem and the frame sequence management subsystem. Since the user-defined command function is mainly concerned with the frame creation, only the frame creation functions are mentioned here.

A graphics tablet (pressure sensitive electronic pad) is adopted as the main data input device. A specially designed authoring sheet, shown in Fig. 1, is set on the graphics tablet and used to create frames on the screen. This sheet consists of the command menu area and the canvas area. The command menu includes basic graphics commands, such as a line, a circle and free-hand; editing commands, such as relocation, reduction and magnification; effect commands, such as painting, flashing, animation, pacing control and timing control. "User-defined command" is also an element of the menu. The canvas area corresponds to the screen. A cursor appears and moves on the screen following the motion of the stylus pen on the canvas area.

The general operation involved in drawing graphics is as follows. First, choose the desired command (e.g. line, circle) by indicating the command menu on the sheet with the stylus pen. Second, specify characteristic points peculiar to the figure, on the canvas area, to decide the location, size and shape. To draw a circle, for example, specify the center and a point on the circle. Screen layout editing can also be accomplished easily, like cutting and pasting written text data.

Most lessons can be created by the LESSONWRITER authoring system without the help of the user-defined command function. However, the user-defined command function enables authors to use advanced techniques in their lessons.

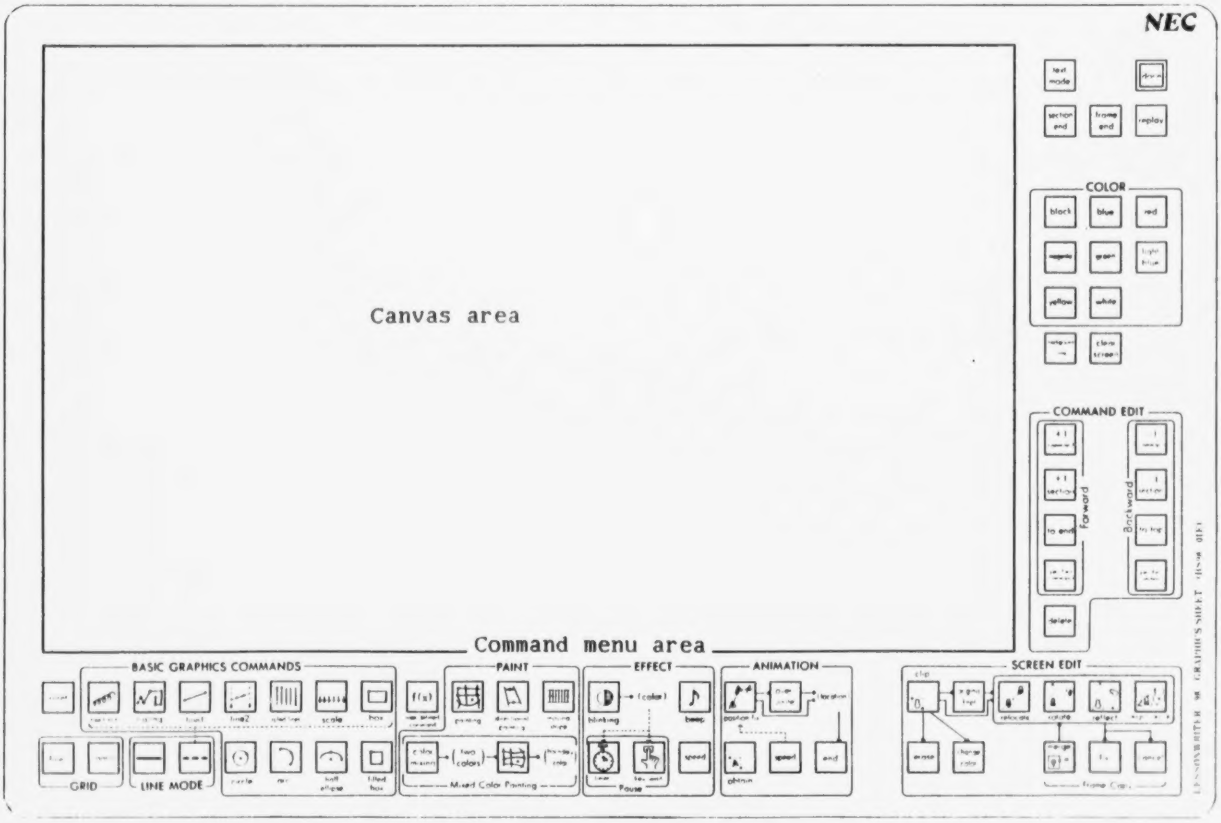

NEC
Canvas area
Command menu area
text mode
section end
frame end
replay
COLOR
black
blue
red
green
light blue
yellow
white
clear screen
COMMAND EDIT
Forward
Backward
to end
to top
delete
BASIC GRAPHICS COMMANDS
line2
scale
box
circle
half ellipse
filled box
GRID
LINE MODE
f(x)
PAINT
painting
two colors
Mixed Color Painting
EFFECT
blinking
(color)
beep
speed
Pause
ANIMATION
obtain
speed
end
SCREEN EDIT
clip
erase
change color
relocate
rotate
reflect
merge

Fig. 1 LESSONWRITER authoring sheet

4. User-defined Command

(1) User-Defined Command Creation

User-defined commands can be written in BASIC languages and registered in the LESSONWRITER authoring system using the utility program. Figure 2 shows a simple example of user-defined command used to draw "a box in perspective" by specifying four points. Various kinds of boxes in perspective can be drawn according to the specified points. This command is realized by the following computer program written in BASIC language.

```
60000   *QSTART
60010   LINE(QX1, QY1)-(QX2, QY2),,B
60020   LINE(QX3, QY3)-(QX4, QY4),,B
60030   LINE(QX1, QY1)-(QX3, QY3)
60040   LINE(QX2, QY1)-(QX4, QY3)
60050   LINE(QX1, QY2)-(QX3, QY4)
60060   LINE(QX2, QY2)-(QX4, QY4)
65000   GOTO *QEND
```

This program draws two boxes and four lines, assuming that the values of variables (QX1, QY1), (QX2, QY2), (QX3, QY3) and (QX4, QY4) are determined. These variables corresponds to the x, y coordinates for the four characteristic points to be specified. The coordinates are automatically read into the variables, when specifying these points on the graphics tablet in the authoring mode. Thus, no elaboration about coordinates digitizing is required in computer programming for user-defined commands.

Sixty commands can be registered in the LESSONWRITER authoring system. It is possible to copy user-defined commands created by other authors. Any of them can be selected from the menu on the display, at the authoring time. The operation involved in specifying characteristic points is the same as for specifying preset graphics commands, such as a line and a circle.

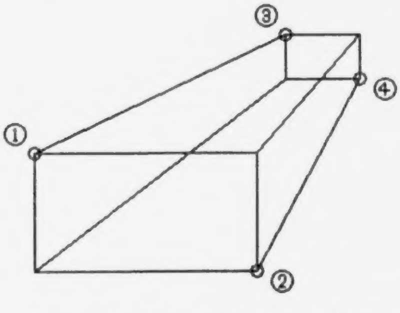
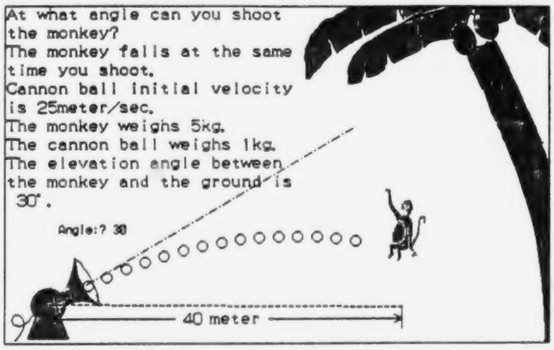


Fig. 2 A box in perspective drawn by a user-defined command
(Numbers indicate the input order for specifying points.)

(2) User-Defined Command Usage Example

The usage of the user-defined command function is explained through an example. Figure 3a is a scene in the frame for explaining the concept of the gravity drop in dynamics through the "Monkey-Hunting Game." A simulation technique is used, so that learners are motivated and able to find the principle through trial and error methods. The goal is shooting down the falling monkey with the cannon ball. Learners can freely change the shooting angle. Complicated background graphics are drawn by a combination of preset graphics commands, including the free-hand command.



At what angle can you shoot the monkey?
The monkey falls at the same time you shoot.
Cannon ball initial velocity is 25meter/sec.
The monkey weighs 5kg.
The cannon ball weighs 1kg.
The elevation angle between the monkey and the ground is 30°.

Angle:? 30

Fig. 3a A "Monkey-hunting Game" scene

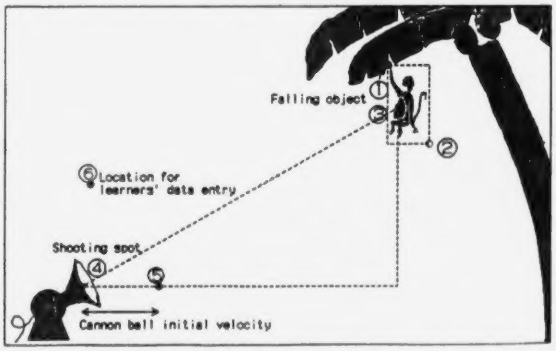

Falling object
①
③
②
⑥ Location for learners' data entry
Shooting spot
④
⑤
Cannon ball initial velocity

Fig. 3b Points to be specified for the user-defined command on the "Monkey-hunting Game"
(Numbers indicate the input order for specifying points.)
①②③: Falling object
④: Shooting spot
④-⑤: Cannon ball initial velocity
⑥: Location for learners' data entry

On the other hand, the help of BASIC language is needed to realize the interactive environment, which includes learners' data entry and the corresponding movement of the cannon ball and the falling monkey. A user-defined command is used. Six points must be specified as parameters to give the BASIC language subroutine the information about the location of the monkey, the shooting spot, the initial velocity of the cannon ball and the location of learners' data entry, as shown in Fig. 3b. Since learners' data entry is processed inside the BASIC language subroutine, independent from the authoring system, learners can enter data at the learning time. Once this command is created, either by the author or by programmers, even when the author wants to change the shape or location of the falling object, or when he wants to change the shooting velocity, the same user-defined command can be used on the authoring system.

(3) User-Defined Command Applications

In addition to the application to simulation lessons, the user-defined command function can be applied widely.

a. Drawing Mathematical Function Curves

Mathematical function curves, such as sine curves and parabolas, often have to be drawn precisely in some subjects, such as mathematics and physics. These curves are usually not available as preset graphics commands in authoring systems and are difficult to draw free-hand. Registering these mathematical function curves as user-defined commands, authors can set up the curves with a few touches, as shown in Fig. 4.

b. Positioning Figures by Calculation

Sometimes, spatial relations between figures have to be expressed precisely. A triangle and its inscribed circle, for example, are made up of lines and a circle, which are commonly available as preset graphics commands in authoring systems. The location of the center of the circle, however, is hard to specify without complicated geometrical procedures or calculation. Employing the user-defined command, merely three-point touch can set up a triangle and its inscribed circle, as shown in Fig. 5.

c. Animation

In addition to the simple animation generating function supported by LESSONWRITER, animation with smooth movement, such as a ball rolling along, can be realized by a user-defined command, using the built-in high speed graphics subroutines.

d. Connection to External Input Devices

User-defined commands enable students to use external input devices, such as a mouse, a graphics tablet and a touch panel as well as a keyboard. This function facilitates students' active participation in learning.

5. Conclusion

The concept and usage of the user-defined commands has been described. This function serves as a user friendly interface between authoring systems and programming languages. The parts of a lesson which require the help of programming languages can be made into modules. Once they are completed and registered, they can be used as pen-

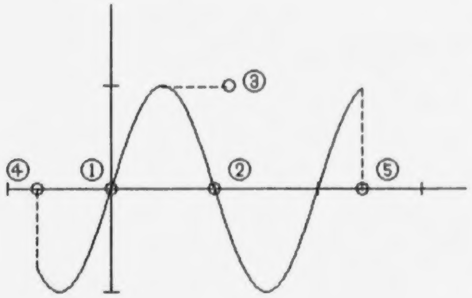


Fig. 4 A sine curve
(Numbers indicate the input order for specifying points.)

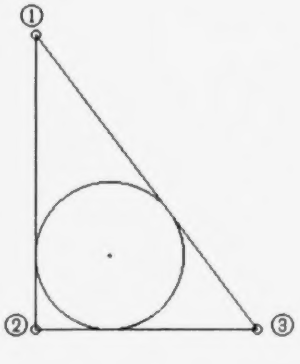


Fig. 5 A triangle and its inscribed circle
(Numbers indicate the input order for specifying points.)

touching commands on the authoring systems. Thus, the authors can develop CAI lessons which require calculation and variable use, such as simulation, forgetting that some parts have been made using a programming language. The user-defined command concept can be greatly contributed to the expansibility of authoring systems.

References

Bork, A. (1985). Computers and Information Technology as a Learning Aid. Education & Computing, Vol. 1, No. 1, pp.25-35.

Chimura, H., Kurata, M., Morimoto, Y. and Sato, T. (1985). A Graphic Teaching-Material Production System Using Personal Computers. Journal of Information Processing, Vo. 8, No. 2, pp.85-92.

Fairweather, P.G. and O'Neal, A.F. (1984). The Impact of Advanced Authoring Systems on CAI Productivity. Journal of Computer-Based Instruction, Vol. 11, No. 3, pp.90-94.

Kearsley, G. (1982). Authoring Systems in Computer Based Education. Communications of the ACM, Vol. 25, No. 7, pp. 429-437.

Kurata, M., Chimura, H., Morimoto, Y. and Sato, T. (1986). A Micro-computer-based CAI Teaching Materials Production System. Proceedings of 27th International ADCIS Conference, pp.78-83.

Morimoto, Y., Chimura, H., Kurata, M., Kato, H. and Sato, T. (1986). Microcomputer CAI Authoring System - LESSONWRITER -. NEC RESEARCH & DEVELOPMENT, No. 80, pp.19-27.

Contact: Yasuhiro Morimoto
Information Basic Research Laboratory
NEC Corporation
1-1, Miyazaki 4-chome
Miyamae-ku, Kawasaki
Kanagawa 213, JAPAN

044-855-1111, Ext. 3431

Implementation and Evaluation of an Original Psychology Software Series

Robert S. Slotnick, Ph. D.
and
Abraham M. Jeger, Ph.D.

New York Institute of Technology

Abstract

An original courseware package, PsychWare, for introductory psychology was developed and field-tested at New York Institute of Technology. PsychWare consists of 10 software programs, a faculty manual, and a student workbook. All programs utilize color, graphics, text and animation. The package personalizes student learning and increases faculty instructional strategies. Field-test results provided the necessary feedback to guide revisions of the programs. A student attitudinal analysis is presented for selected programs.

During the past 3 years the New York Institute of Technology has developed an original courseware package for introductory psychology (Slotnick, 1986). This instructional package, called PsychWare, consists of 10 psychology software programs, an accompanying faculty manual, and a seperate student workbook. The software programs are an interactive, series of simulations, experiments and tutorials on selected course topics. The materials can serve as stand-alone exercises in a microcomputer laboratory, be used as demonstrations in lectures and supplement standard text presentations. The purpose of this paper is to describe PsychWare and its innovative application for teaching psychology.

Overview of PsychWare: The Instructional Package

PsychWare represents the first wave of simulation-based software for introductory psychology. The software exploits the symbolic construction and manipulation powers of the computer in addition to its usual number crunching power. This series builds on earlier software programs developed for psychology and makes use of suggestions appearing in the general educational technology literature. Our work exemplifies some of the suggestions for applying computers to the teaching of psychology called for by Collyer (1984)--namely, "emphasizing the computer as an object-to-think-with rather than simply as an object-that-thinks" (p.206).

Through our software programs, the computer supplements and enhances standard text material in several important ways. Multiple modalities of computer enhanced learning (CEL) are reflected in various prgrams, e.g., simulations and tutorials, gaming, and "student-as-subject" in experiments. All programs convey information to the student in multimodal sensory channels by combining color, graphics, text, and animation. Program design and screen layout utilize all these features to enhance presentation of materials. Further, the programs are highly interactive requiring intelligent student responses in order to move sequentially through the various components to complete the lessons. The software series seeks to reinforce the understanding of classic experiments and basic psychology principles through experiential learning.

The comprehensive educational package, with faculty and student guides, serves a number of pedagogical goals. It aims to make complex concepts more concrete and manageable for the students. At the same time, the modality encourages students to pay attention to specific detail when answering questions and acquiring basic concepts. The hands-on approach gives students an awareness of the process involved in analyzing data and making inferences. And perhaps, most importantly, learning the material in an experiential and interactive manner deeply personalizes the learning process. Teaching students to use simulations, which include setting up experiments, testing hypotheses, manipulating variables, and experiencing the results directly, gives the faculty powerful new instructional strategies.

Faculty Resource Guide and Student Workbook

To integrate the software programs with the course contents (lecture and text material) substantive print support materials were developed. A faculty resource guide for each program contains learning objectives, discussion questions, and guidelines for evaluating student responses. The faculty guide is fully illustrated and includes instructions, pictures, tables and graphs, frequently, as they appear in the programs. This enables faculty to follow the program sequence and score student responses easily.

A student workbook contains laboratory worksheets for recording, graphing, analyzing, and making inferences from the data. Questions follow the program sequence closely which helps students focus on key points. In addition, questions probe generalization and discrimination of the concepts demonstrated in the programs.

Clear instructions to faculty and students on how to operate the computer and interact with the software programs are included in the print manuals. The programs are self-booting (starting) and menu driven.

Software Description

The following are brief summaries of the individual programs that constitute this software series. All programs run on the Apple II series microcomputers.

1. Operant Conditioning I

This interactive program introduces the basic principles of operant conditioning through simulation of various experiments in learning. The experiments are conducted in a simulated Skinner Box enviorment, with a pigeon, an illustrative cumulative plotter, a stimulus key and food hopper. The student conditions the pigeon to a continuous reinforcement (CRF) schedule. In the CRF module a game-like feature is introduced by comparing the accuracy of the student's reinforcements with the actual number.

2. Operant Conditioning II

In this part of the program the student learns the effects of partial reinforcement schedules on acquisition and extinction. The student then assumes the role of the pigeon to learn a discrimination task. In the final module the student controls the pigeon's behaviors while learning a behavior chaining task.

3. Incremental Learning: A Self-Experiment

The student as subject in an experiment. His/her data are tabulated, graphed and compared to normative data. Incremental learning is demonstrated on a specific learning task. The student participates in data collection and observes its transformation from raw data to graphic modes.

4. Classical Conditioning

This is a tutorial program with nested simulations. It teaches the basic principles of classical conditioning through the presentation of color graphs, animation, symbolic representations and simple clear test. The student must complete several simulations to move from acquisition of conditioning to extinction to generalization and discrimination.

5. Digit-Span

The student serves as subject in a memory experiment; his/her data are tabulated and graphed. In the first experiment a series of digits are presented one at a time; in the second experiment the digits are presented in chunks of 3. This program is designed to help students discover the importance of chunking in short-term memory. Chunking is compared to non-chunking conditions using color graphs.

6. Prisoner's Dilemma

This program demonstrates the consequences of decision strategies involving trust vs. suspicion in a simulated crime situation. Students assume the role of one of the prisoners and play against the computer, which is programmed to simulate a second prisoner. The outcomes are combinations of jail sentences or acquitals. Analysis of the results provide students with feedback as to the decision strategy they adopted.

7. Gambler's Dilemma

This program utilizes the strategies of the Prisoner's Dilemma in a 2-person interactive game. Cooperation and competition are the conflicting motives that the students must balance in trying to accumulate points (win "money"). The program tracks responses and outcomes and provides an analysis of the students' strategies. There are three different game matrices, each with a different reward structure. Students can analyze their strategies as a function of the differential rewards.

8. Memory: Words Vs. Imagery

The student is asked to memorize a list of word pairs and a list of image pairs. One word of each word pair (or one image of each image pair) is then presented and recognition of the second word or image is tested. A graph of results is presented. The student is asked to repeat the experiment with a different list of words and set of images. The program features very attractive graphics for the images and the graphs.

9. Cognitive Development

Piaget's four stages of cognitive development are taught in a tutorial and simulation format. The student first learns the major characteristics of the four stages and then applies developmental tests, e.g., conservation of identity test, seriation test, and the pendulum problem, to a computer simulated child. Based on the simulated results the student must assign the child's cognitive stage.

10. Short Term Memory

In this program the student serves as a subject in a number of short term memory experiments. The experiments include estimation of dots, memorization of letters, syllables, and words. The concepts illutrated are the roles of chunking and chunking-plus-meaning in organinzing stimulus materials. Student data are collected and graphed for four experiments.

Field Testing

During the software development process individual exercises were subjected to field testing with introductory psychology students. Field test results provided the necessary feedback to guide revisions of the programs.

PsychWare exercises were field tested in the fall, 1984 and spring, 1985 semesters at the New York Institute of Technology, Metropolitan Campus. In lieu of a classroom lecture the professor conducted a laboratory session at the Microcomputer Laboratory. During the laboratory hour students interacted with tutorial and simulation exercises, which provided students with an opportunity for hands-on experience with this new learning modality. Following the computer interaction, students were required to complete a laboratory worksheet geared to the particular exercise. This represented the only opportunity for students in this course to engage in writing for purposes of synthesizing and integating acquired concepts.

An analysis of the student lab sheets revealed that, in most cases, they understood the computer exercises. They demonstrated knowledge of the content-specific questions, showed mastery of general principles, and grasped the underlying concepts.

The lab exercises and accompanying worksheets provide early feedback to the instructor as to which aspects of the subject matter are especially troublesome to students. This served to guide reviews of difficult concepts in subsequent class sessions prior to an exam.

Another component of our field testing involved subjecting the software programs to student attitudinal analysis. An "Attitudes Toward Microcomputer Exercise" questionnaire was developed for this purpose. It consisted of 4 items, each rated along a 5-point scale. Questions focused on the extent to which students perceived the computer exercise as a positive learning experience, the extent to which the computer program helped students understand the subject matter, the ease of instructions, and the extent to which computer learning was preferred to textbook learning. In addition, an open ended item was included, requesting students to suggest ways in which the computer exercise can be improved.

Table 1 presents scores on this questionnaire, including means for individual scale items and overall scale mean, for two selected software programs--Classical Conditioning and Gambler's Dilemma. These data are presented to illustrate the field testing/revision process. With full implementation of the final version, such data can be collected to determine between group attitudinal differences, as well as changes over time following Faculty experience in a particular institution.

As indicated in the table, student perceptions are positive, with an overall mean score of 16.20 and 14.65 for the Classical Conditioning and Gambler's Dilemma, respectively. Within scale item variability was narrow for the Classical Conditioning program with wider variations apparent in the Gambler's Dilemma exercise. This is probably due to the fact that Classical Conditioning is a tutorial and simulation while Gambler's Dilemma is a more complex 2-person game which requires more clarification. Indeed, based on students' and others' feedback, changes were made following this semester's experience to facilitate the learning objectives and ease-of-use of the program.

Conclusion

These interactive programs with accompanying print materials offer an exciting new educational delivery modality. The software programs are characterized by the following unique features:

.high level of interaction

.full use of color and graphics

.animated sequences

.experiential learning

.teach to clear educational objectives

.include faculty resource manual; and,

.a complete student workbook.

Large and often impersonal lectures can be supplemented with more intensive and experiential learning. Students serve as both experiementers and subjects in experiments and write lab reports on these experiences--bringing authenticity and impact to the teaching of psychology.

References

Collyer, C.E. Using computers in the teaching of psychology: Five things that seem to work. Teaching of Psycholgy, 1984,11, 206-209.

Jeger, A.M., Slotnick, R.S., Thomas, J., Kohn, M. and Peng, D. Evaluation of the Title III Pilot Inplementation of CAI in Math and English Courses: The 1983-84 experience. New York Institute of Technology, N.Y., N.Y., 1984.

Slotnick, R.S. PsychWare An Introductory psychology software series. Minneapolis, Mn.: West Publishing Co., 1986.

Notes

1. This project, Technology Based Education, has been partially suported by U.S. Dept. of Education Grant, # G008202189, to NYIT.

TABLE 1

MEANS ON "ATTITUDES TOWARD MICROCOMPUTER EXERCISE" QUESTIONNAIRE FOR TWO PSYCHOLOGY PROGRAMS (INDIVIDUAL ITEMS AND TOTAL SCALE SCORES).

	Means	
TITLE OF PROGRAM:	Classical Conditioning	Gambler's Dilemma
* ITEMS:	(N=14)	(N=17)
a. Positive Learning Experience	4.00	3.18
b. Understanding Subject Matter	3.85	3.71
c. Ease of Instructions	4.14	4.29
d. Preference Over Textbook Learning	4.21	3.47
** Overall Scale Means :	16.20	14.65

---

* Individual items range from 1-5, with 5 being the most favorable response.

** Scale scores range from 4-20.

ADCIS Proceedings, November 1986

KAPLEX: A Software Architecture for Knowledge-Based Authoring Systems

Y. Richard Wang
Jay F. Nunamaker
Minder Chen

Department of Management Information Systems
College of Business and Public Administration
University of Arizona, Tucson, 85721

ABSTRACT

This paper studies the evolution and enhancement of knowledge-based authoring systems (KBASs). Conventional authoring systems and intelligent tutoring systems are reviewed, their functionalities are summarized, and a software architecture for KBASs, called KAPLEX, is proposed. KAPLEX integrates conventional authoring systems and intelligent tutoring systems under a cohesive architecture which encompasses authoring management, courseware management, and delivery management. The authoring advisor, access manager, private instructor, and other supporting modules are identified. Finally, research directions on KAPLEX are presented.

1 MOTIVATION AND BACKGROUND

An authoring system (AS) [Kearsley, 1982] is a program or a set of programs which allows an instructor or instructional developer (the author) to create courseware without programming. A knowledge-based authoring system (KBAS) integrates theories and models of instructional design, authoring tools, knowledge engineering, tutoring strategies, and student modeling to form the bases for all interactions among the author, the system, and the student. It provides an environment and facilities for the efficient development and delivery of courseware in different subject matters.

The purpose of this paper is to trace the evolution of KBASs, to identify research issues and trends, and to propose a system architecture for KBASs. Three activities are identified in a KBAS: authoring management (knowledge acquisition), courseware management (knowledge storage), and delivery management (knowledge transfer). The application of software engineering strategies permits researchers to analyze the complexity of KBASs without losing sight of the details of their components. To present the paper concisely, the architecture proposed by the authors is called "KAPLEX".

2 EVOLUTION OF AUTHORING SYSTEMS

VAULT [Romaniuk, 1970] was one of the early authoring systems in which the instructional logic was separated from the content during the authoring process. This concept was further evolved in the TICCIT system [Merrill, 1980] in which the learner control was built into the system and the author simply provided the content. Many authoring systems were developed for the PLATO system [Kearsley, 1982]. The availability of microcomputer has led to development of authoring systems such as PCD1, EDUCATOR, and AUTHORITY. None of these conventional authoring systems (CASs), whether designed for the mainframe or the micro, is capable of individualizing instruction in any profound manner, nor is any of them capable of sophisticated diagnosis of student errors [Barr and Feigenbaum, 1982; Kearsley, 1982].

A different approach to the "conventional" system is AI-based Intelligent Tutoring Systems (ITSs) [Carbonell, 1970; Koffman and Blount, 1975; Brown, 1977; Gable and Page, 1980; Sleeman and Brown, 1982; O'Shea et al., 1984]. Early research on ITSs focused on representation of the subject matter; as did SCHOLAR on South American geography, SOPHIE on circuit trouble-shooting, and EXCHECK on logic and set theory. The high level of domain expertise in these programs permits them to respond to a wide range of problem-solving interactions, but they are not necessarily good teachers of the material. To rectify this deficiency, tutoring strategies and models of the student's learning behavior have been proposed [Goldstein, 1977; Burton and Brown 1979; Clancey, 1979; Barr and Feigenbaum, 1982]. However, these ITSs are ad hoc in nature, focused on tutoring strategies and student models instead of the overall authoring and delivering processes, and each program represents a major investment of resources. Sleeman and Brown [1982] pointed out that:

> "ITS has clearly abandoned one of CAI's early objectives, namely that of providing total courses, and has concentrated on building systems which provide supportive environments for more limited topics. If ITSs are to have any practical importance, the subject areas need to be chosen with considerable care as each system represents a major investment of resources... (the authors are) expecting more ambitious and generalizable ITSs to emerge in the next few years."

It is interesting to observe that fifteen years of research in authoring systems has resulted in two schools of systems: On the one hand, CASs emphasized ease of use and cost-effectiveness but overlooked student modeling and tutoring strategies. On the other hand, ITSs sought individualization and diagnosis but neglected authoring tools.

Some of the ITSs offer domain independence, explicitly or implicitly, by changing contents in the knowledge and data bases. For instance, SCHOLAR applied semantic networks to represent South American geography. This semantic network approach has also been applied, in NLS-SCHOLAR, to the text-editor domain [Barr and Feigenbaum, 1982]. Similarly, the procedural knowledge studied in BUGGY has been applied to several different problem domains to find students' systematic errors. In GUIDON, the inference engine and knowledge base were explicitly decomposed to account for domain independence. GUIDON has been implemented on several problem domains such as medical diagnoses and structural analyses. In the advanced computer tutoring project conducted at Carnegie-Mellon University [Anderson et al., 1985], a domain-free theory of instruction is being developed.

The only generic knowledge-based authoring system, to our knowledge, was proposed in by O'Shea et al. [1984] and is one in which production rules were used to facilitate the "mixed initiative" dialog. The focus of that paper was on the creation of intelligent tutors, using a simple five-component model. The work by O'Shea et al. and Anderson et al. echoes what Sleeman and Brown [1982] expected of the more ambitious and generalizable KBASs. In this paper, the authors attempt to integrate CASs and ITSs. Figure 1 depicts the evolution toward KBASs.

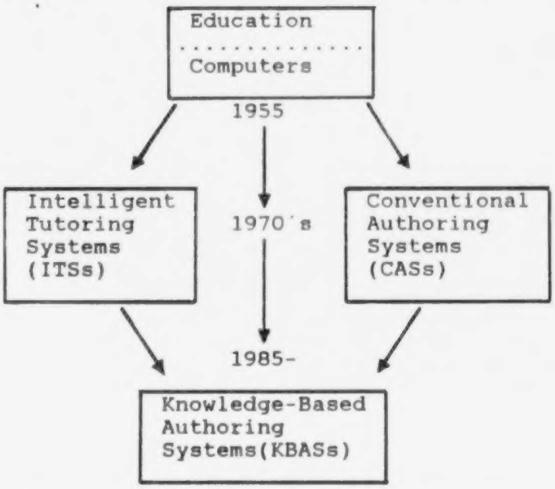

Education
....
Computers
1955
Intelligent Tutoring Systems (ITSs)
1970's
Conventional Authoring Systems (CASs)
1985-
Knowledge-Based Authoring Systems(KBASs)

Figure 1: The evolution toward KBASs

3 FUNCTIONALITIES OF THE KBASS

Based on a literature search [Wang and Wei, 1985], a set of functionalities for CASs is presented in Table 1 which is partitioned into four levels: content creation, lesson definition, course management, and authoring environment [Kearsley, 1982]. CASs were primarily designed to speed up the courseware development process rather than for individualized instruction, sophisticated diagnosis, or remediation of student errors.

Table 1: Functional Specification of CASs

Level	Functionality
Content Creation	Animation, Audio, Color, Graphics, Input Devices, Mode, Text, Video
Lesson Definition	Branching Definition, Calculation, Display Creation, Display Utility, Function Key, Looping, Slide, Student Input Processing, Subroutine Generation, Viewport (Windows)
Course Management	Control Options for Rate & Sequence of Lesson, Data Base, Instructional Mode, Management of Student Achievement, Random Number Generator, System Mode,Testing Mode
Authoring Environment	Utilities, Communication, Data Structure Function Calls, Help, Multi-functions of Learning, Multi-level of Hardware Configuration, Needs Assessment, Productivity, Software Portability, Training, Type of Lesson Creation.

Most ITSs have tutoring strategies and student models built into the system to monitor a student's progress and then to try to find the best way to teach that individual. Tutoring strategies used in the existing ITSs include: computer coach [Burton and Brown, 1982], learning by doing [Sleeman, 1983], game playing [Barr and Feigenbaum, 1982], Socratic tutoring heuristics [Barr and Feigenbaum, 1982], tutorial [Buchanan and Shortliffe, 1984], ACT learning theory [Reiser, Anderson, and Farrell, 1985], case method [Buchanan and Shortliffe, 1984]. Current ITSs support only one or two tutoring strategies, which raises the following issues:

1. How can multiple tutoring strategies be built into one system?
2. What are the relationships among student models, tutoring strategies, and domain knowledge?

Two student models have been widely used in the existing ITS literature:

1. Overlay Model [Goldstein, 1982],
2. BUGGY Model [Burton, 1982].

Modeling the student's behavior is among the most difficult tasks for even a skilled human teacher [Goldstein, 1982]. A good student model should incorporate solutions to the following questions:

1. How do students represent the domain knowledge in their mind?
2. How can a student model be used to find out students' misconceptions and bugs?
3. What is the best tutoring strategy and presentation method, given the current state of knowledge in the student model?
4. How can a student model be updated based upon the analysis of a student's answers and learning behavior?
5. Which parts of a student model depend upon the domain knowledge?

We have summarized the functionalities of CASs and ITSs in terms of content creation, lesson definition, course management, authoring environment, tutoring strategies, student modeling, and knowledge representation. It is found that the functionalities of CASs and ITSs are intertwined. Moreover, a number of aspects of KBASs are not studied in either CASs or ITSs. For instance, autostoryboarding (the mechanism provided by the system to facilitate the storyboarding process including idea parking, brainstorming, and story editing) is an important aspect of the authoring process which is not discussed in the literature. Autostoryboarding is handled manually in the existing CASs and hardwired by the authors of ITSs. Another important aspect of KBASs which is only beginning to be addressed is the representation of knowledge in a uniform internal structure and the corresponding conversion to different external views. In sum, it is a complicated task to decompose all the intertwining functionalities cleanly and clearly. This issue is addressed by answering another fundamental question: what is a KBAS supposed to do?

By definition, there are three activities involved: authoring management (knowledge acquisition), courseware management (knowledge storage), and delivery management (knowledge transfer). The authoring management component can be further decomposed into instructional strategies, an authoring advisor, and authoring tools. The courseware management component can be decomposed into representation strategies, an access manager, and knowledge and data. Similarly, the delivery management component can be decomposed into tutoring strategies, a private instructor, and student models. A high-performance, high-capacity knowledge and data base computer, such as INFOPLEX [Madnick, 1979], can be applied to support the activities involved in the KBAS. Figure 2 depicts the architecture of KAPLEX, which includes these nine modules. Each module in the system interfaces with others through well-defined I/O functions as discussed below. Note that the labels "(A)" through "(H)" correspond to the letters "A" through "H" in Figure 2.

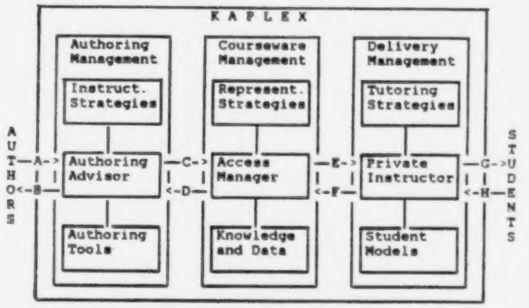


Figure 2: The Architecture for KAPLEX

(A): New knowledge and modification to knowledge and data base from authors; for example, new rules and facts.

(B): Explanation, verification, consultation, and feedback to authors; for example, explanation of usage of the KBAS, verification of completeness and consistency of knowledge and information entered, consultation of instructional strategies, and feedback of students' performance.

(C): New knowledge and modification to knowledge and data base in different knowledge representation schemes.

(D): Knowledge and information retrieved from knowledge and data base for functions in (B).

(E): Knowledge and information retrieved from knowledge and data base for functions in (G).

(F): New input and modification from the student to knowledge and data base including basic data, learning style, and performance data.

(G): Customization and delivery of courseware in different presentation mode to students; for example, customization of courseware using the domain knowledge, tutoring strategies, and the student's state of knowledge; delivery of courseware using instructional design theories and the student's learning style.

(H): Queries and responses from students.

4 DISCUSSION AND FUTURE DIRECTIONS

We have presented the architecture of KAPLEX. Unlike other architectures [O'Shea et al., 1984; Kearsley, 1982; Buchanan and Shortliffe, 1984], which were designed to speed up the courseware development or to focus on the reactive learning environment, KAPLEX encompasses authoring management, courseware management, and delivery management. It permits us to clearly envision the interaction among the modules. The streamlined information flow among the modules in KAPLEX also helps the design process. Most importantly, the architecture of KAPLEX provides us with an intellectual foundation to analyze theories and models of authoring systems, instructional models, knowledge representation schemes, tutoring models, and student models without being diverted by the intricacies of any of the disciplines.

The following tasks, among others, are being performed to continue the research:

1) a critical review and a systematic survey of theories and models to further specify and substantiate KAPLEX,
2) design of an intelligent computer structure tutor,
3) conversion of knowledge and data from multiple external views to a uniform internal structure, and
4) autostoryboarding.

Research issues will surface as we continue to apply software engineering techniques and conduct case studies to substantiate and to validate the KAPLEX architecture. It will pave the way for the successful implementation of KAPLEX.

5 ACKNOWLEDGEMENTS

The authors wish to thank Irene Chen and Bill Saints for their assistance in writing this paper. Work reported herein was supported, in part, by a grant from the NCR corporation.

6 REFERENCES

Anderson, J. R., Boyle, F. C., & Yost, G. The geometry tutor. *Proceedings of the Ninth International Joint Conference on Artificial Intelligence, IJCAI-85*, 1985, Volume 1, 1-7.

Barr, A. & Feigenbaum, E. A. (Eds.). (1981-1982). *The handbook of artificial intelligence* (Vols. 1-2). Los Altos, CA: William Kaufmann.

Brown, J. S. (1977). Use of artificial intelligence and advanced computer technology in education. In R. J. Seidel & M. Rubin (Eds.), *Computers and communications: Implications for education*. New York: Academic Press, 1977.

Brown, J. S. & Burton, R. R. A paradigmatic example of an artificial intelligent instructional system. *International Journal of Man-Machine Studies*, 1978, 10, 323-439.

Buchanan, B. G. & Shortliffe, E. H. (Eds.). *Rule-based expert systems: the MYCIN experiments of the Stanford heuristic programming project*. Reading, MA: Addison-Wesley, 1984.

Burton, R. R. Diagnosing bugs in a simple procedural skill. In D. Sleeman & J. S. Brown (Eds.), *Intelligent tutoring systems*. New York: Academic Press, 1982.

Burton, R. R., & Brown, J. S. An investigation of computer coaching for informal learning activities. *International Journal of Man-Machine Studies*, 1979, 11, 5-24.

Carbonell, J. R. AI in CAI: an artificial intelligence approach to computer-aided instruction. *IEEE Transactions on Man-Machine Systems* MMS-11, 1970, 4, 190-202.

Clancey, W. J. Transfer of rule-based expertise through a tutorial dialogue. (Doctoral dissertation, Stanford University, 1979).

Gable, A. & Page, C. V. The use of artificial intelligence techniques in computer-assisted instruction: an overview. *International Journal Man-Machine Studies*, 1980, 12, 259-282.

Goldstein, I. *The computer as coach: an athletic paradigm for intellectual education*. Cambridge, MA: AI Laboratory, MIT, 1977.

Goldstein, I. The genetic graph: a representation for the evolution of procedural knowledge. In D. Sleeman & J. S. Brown (Eds.), *Intelligent tutoring systems*. New York: Academic Press, 1982.

Kearsley, G. Authoring systems in computer based education. *Communication of the ACM*, 1982, 25, 429-437.

Koffman, E. B. & Blount, S. E. Artificial intelligence and automatic programming in CAI. *Artificial Intelligence*, 1975, 6, 215-234.

Madnick, S. E. The INFOPLEX database computer, concepts and directions. *Proc. IEEE Comp. Con.* (February 1979): 168-176.

Merrill, M. D., Schneider, E., and Fletcher, K. *TICCIT* Englewood Cliffs, NJ: Educational Technology Publications, 1980.

O'Shea, T., Bornat, R., Du Boulay, B., and Eisenstadt, M. "Tools for Creating Intelligent Computer Tutors. in Artificial and Human Intelligence. *Proceeding of the International NATO Symposium* (Lyon, France, Oct. 1981). Edited by A. Elithorn and R. Banerji, Elsevier North-holland, Inc., New York, NY, 1984, 181-199.

Reiser, Brian J., Anderson, John, R., and Farrell, Robert G. Dynamic Student Modelling in an Intelligent Tutor for LISP Programming. *Proceedings of the Ninth International Joint Conference on Artificial Intelligence, IJCAI-1985*. Volume 1, pp. 8-14. UCLA, 1985.

Romaniuk, E. W. *A Versatile Authoring Language for Teachers* Ph. D. dissertation, University of Alberta, Edmonton, Alberta, 1970.

Sleeman, D. & Brown, J. S. (Eds.), *Intelligent tutoring systems*. New York: Academic Press, 1982.

Spreadsheets and Databases - Alternatives to Programming

M. Gene Bailey, Ph.D.

East Tennessee State University

ABSTRACT: Microcomputers have become easier and easier to use as an emphasis is placed on software design for users with little or no experience. With a minimal amount of training, students or employees can become productive. Yet, academic programs still teach programming as a fundamental part of computer literacy. This paper proposes that the programming portion of the class be eliminated and replaced with a study of spreadsheets and databases. The paper discusses the manner in which spreadsheet and database design can be approached to maximize learning.

An important concept that is learned in any programming class is algorithm development. Students learn how to analyze the problem and to set up the step-by-step solution. This process must be done before any coding can begin. Spreadsheets offer the same type of learning and are much more appropriate for non-computer science majors. Students must formulate the problem, determine the equations and formulas necessary to solve it, and then set up the worksheet. Spreadsheet applications are found in every discipline and their design requires a minimal amount of computer knowledge and mathematics.

Learning to set up a database and design the queries necessary to obtain information from the database is a concept important to computer literacy. Once again, students learn to "program" by learning to construct their questions in a manner that is meaningful to the software package.

Programming vs. Spreadsheets

Problem analysis is the most important phase in writing a computer program to solve a problem. The actual coding is done only after the step-by-step solution has been determined. The process is akin to a chef who through experimentation has discovered a magnificent recipe. Once written down in a step-by-step form, the recipe can be "coded" in any language desired. The emphasis in the classroom, then, should be placed on giving students the tools they need to do problem analysis. By using a software package which will permit this analysis and also the design and implementation, students can concentrate on the problem solution without being overwhelmed by the idiosyncracies of a computer programming language.

There are three essential programming concepts that are used in computer programming: sequential execution, transfer of execution, and repetition. The first of these simply means that when one instruction has been executed the next one in sequence is executed. The second means that some decision must be made and the instruction(s) to be executed next will depend on the result of that execution. The third means that instructions are repeated over and over. All three of these concepts can be learned by using spreadsheet and database software packages.

The art of spreadsheet design draws heavily, albeit subtly, on the mathematical concepts of the 8th/9th grade algebra. The intersection of a column and row is labelled a "cell" and the name of the cell is determined by the column and row that form it. For example, the name of the location given to the intersection of column C and row 7 is C7. Values or labels are placed in cells, just as values and labels are placed in memory locations using a given programming language convention. Students learn how to construct formulas using the names of cells just as they would learn to construct formulas using the names of locations in a programming language.

Consider the example below where the object is to determine the average of the exams and to find individual student averages for a sequence of exams. This particular problem is a common one with introductory programming classes because the logic involved is one with which all students are familiar. Determining the solution is, then, simply one of using the appropriate tools in the "programming language."

```
         A                 B      C      D     E   F
1.   NAME              TEST1  TEST2  TEST3  SUM  AVG
2.   -----------------------------------------------
3.   Doe, John            76     80     90  ___  ___
4.   Evans, Anne          99     88     82  ___  ___
5.   Frank, June          80     89     76  ___  ___
6.   Mason, Bill          70     80     60  ___  ___
7.   -----------------------------------------------
8.   Test Avg:           ___    ___    ___
```

The letters over the columns and numbers on the rows correspond to the lettered columns and numbered rows of a spreadsheet. Students are initially instructed to write down exactly what they would do if they were to calculate an average test grade for John Doe. The solution is a two-step one; first, find the sum of the grades, and then divide by 3. The sum of the grades is

76+80+90

They are, then, instructed to replace the values with the names of the cell that contains the respective value. A "plus sign" must precede the first cell name so that the spreadsheet interprets the expression as an equation and not as a label. In this case, the equation would read:

+B1+C1+D1

This, then, is the entry made for Cell E1. As soon as the formula is entered, the computer displays 246 on the screen. It is made clear to the students that the computer has really stored the formula and is only "displaying" the answer. Thus, changing a test grade results in a different sum. The corresponding equation written in BASIC would be

E1=B1+C1+D1

The spreadsheet and programming language statements are identical, but much less time is spent in explaining "how" values are placed "into" B1, C1, and D1. One further advantage is the immediate feedback that result when the formula is entered into E1. Students will know instantly whether their formula is correct. The column of averages is obtained in a similar manner:

```
246/3  or  +E1/3  is entered into F1
           +E2/3  is entered into F2,
           etc.
```

the corresponding formulas in BASIC would be F1=E2/3, F2=E2/3, etc.

Once students have grasped this simple construction technique for finding the sum of the test scores the SUM column is deleted and formulas for the averages are calculated directly using appropriate parentheses. The formula entered into the new E1 is

```
         A              B       C       D      E
1.   NAME           TEST1   TEST2   TEST3    AVG
2.   --------------------------------------------
3.   Doe, John         76      80      90    ___
4.   Evans, Anne       99      88      82    ___
5.   Frank, June       80      89      76    ___
6.   Mason, Bill       70      80      60    ___
7.   --------------------------------------------
8.   Test Avg:        ___     ___     ___
```

(B1+C1+D1)/3 and in BASIC :
E1=(B1+C1+D1)/3

The students complete the spreadsheet and determine averages for each exam as well as for each student. The problem is then changed to reflect different percentages for the exams. For example, suppose that each of the exams 1, 2, and 3, had 30%, 20% and 50% weights assigned to them. The objective is, again, to find the average. The first step is to write down the solution using numbers:

.3*76+.2*80+.5*90

and, then, to replace any number corresponding to the contents of a cell with the cell name:

.3*B1+.2*C1+.5*C1

The resulting equation is placed into cell E1. The corresponding equation in BASIC would read:

E1=.3*B1+.2*C1+.5*D1

Once students have learned to construct their own average and weighted average solutions, they examine the built-in functions @SUM, @COUNT, and

@AVG and modify their formulas to use these. For example, the formula (in E1) to find John Doe's test average could either be

```
(+B1+C1+D1)/3  or
@SUM(B1..D1)/3  or
@SUM(B1..D1)/@COUNT(B1..D1)  or
@AVG(B1..D1)
```

The printing process is so much more easily defined and explained when using a packaged program. Students simply have to learn what parameters to provide to specify certain ranges, headings, borders, etc. Once again, an often confusing part of programming (for beginners) is determining how to print desired results and to construct the appropriate programming language statements to do it for you. With spreadsheets, students still have to determine WHAT to print; the HOW is handled by the package.

Another example used to help students to learn problem solving is the ever-popular "payroll" problem. This particular programming exercise is probably assigned in every introductory computer programming course. The reason is obvious--everyone understands the logic involved in calculating pay, deductions, and net salary. Therefore, students are free to concentrate on the language, i.e., formulating the solution in a form the spreadsheet can use.

The spreadsheet for this example would resemble the FIGURE 1 at the end of this paper.

To make the problem simple, the percentages for FICA and FED TAX could be constants, say 7% and 15% respectively. This particular problem introduces the decision concept important to computer programming, which alters the sequence of execution based on some criteria. In this case, if the employee works more than 40 hours, the pay is time and a half.

The student is instructed to write out the solution to the overtime problem using two steps; first, to determine the number of overtime hours and second, the overtime pay. The overtime hours could be found using either of two formulas. First,

maximum of (hours - 40) and 0

Thus, if the hours worked is 35, then the maximum of 35-40 or (-5) and 0 would be 0. The hours worked for the first person is contained in cell C3. The @MAX formula is used and the following expression is entered into cell D3.

```
@MAX(C3-40,0)
```

The second technique for determining the number of overtime hours worked is to use the below logic:

> If the hours worked is greater than 40, then overtime hours are found by subtracting 40 from the hours worked. Otherwise, use the value 0.

The @IF function would be used here to formulate the solution:

```
@IF(C3>40,C3-40,0)
```

This expression would be placed in D3 and means simply that the computer is to compare the contents of cell C3 with 40 and if the value found there is bigger than 40, then the contents to be shown on the screen in cell D3 is the difference. Otherwise, the value on the screen in cell D3 is 0.

This particular exercise is difficult to do in BASIC programming because it requires the student to branch to another location to continue processing. Consider the below BASIC instructions.

```
 90  OVT=0
100  IF HOURS > 40 THEN 120
110  GOTO 130
120  OVT = HOURS - 40
130   processing continues here
```

Another BASIC solution requires students to reformulate their logic and determine if the employee did NOT work overtime. The BASIC code for this would be:

```
 90  OVT=0
100  IF HOURS <= 40 THEN 120
110  OVT = HOURS - 40
120   processing continues here
```

Another difficulty confronting students when using a high-level programming language is the problem of placing the appropriate values into the storage locations OVT and HOURS. It is very clear from the spreadsheet design that these values are actually entered into specific cells. The problem of naming locations is essentially removed by using the name of the cell that contains the value.

The remainder of the exercise, i.e., to determine overtime pay, gross salary, deductions and, ultimately, the net salary is now relatively simple. The formula, below, for calculating overtime pay for the first employee is entered into E3.

```
overtime hours * pay-and-a-half.
   +D3         * 1.5 * B3
```

If the employee had no overtime hours, then the value in D3 would be 0 and overtime pay would be 0. Calculating gross pay requires some thought before writing the formula. The algorithm to use is:

> If the employee worked less than 40 hours, then the gross pay is simply hours worked * pay rate. However, if the employee worked more than 40 hours, the gross pay is the sum of regular pay (40 * payrate) and overtime pay.

The spreadsheet expression for this algorithm is given below and is entered into F3.

```
@IF(C3<=40,B3*C3,40*B3+E3)
```

The deductions and net pay are determined by inserting the below formulas into G3, H3, I3, and J3 respectively.

```
.07*F3  (determine FICA as 7% of
        Gross)
.15*F3  (determine Fed Tax as 15%
        of Gross)
+G3+H3  (determine Total deduction)
+F3-I3  (determine Net Pay after
        deductions)
```

The corresponding BASIC program to find the net salary is much more complicated because of branching and because of the confusion caused in using appropriately named storage locations. Another problem with using a programming language is that the program must be executed before the students know whether their logic is correct. On the other hand, a spreadsheet gives immediate feedback indicating correct/incorrect logic when test values are inserted into the cells in the HOURS and PAY RATE columns.

In each of the examples listed above, only the formulas for the first line of each problem were given. Using the corresponding cells, students could construct the formulas for the remaining lines. Once it is clear that the "concept" of formula building is understood, the instuctor can demonstrate the formula copying process which eliminates a great deal of keyboarding.

Many problems cannot be solved by constructing spreadsheets, but most of the problems discussed in "Computer Literacy" or "Computer as a Tool" classes are of a type much like those problems given above, straight-forward business applications that repeat a process over and over. This particular type of application lends itself, readily, to spreadsheet analysis.

Additional advantages to using spreadsheet include "what if" experimentation. Students can alter values searching for the "best" solution to a problem. The use of a @PMT formula which determines the payment of a loan at a given rate of interest for a certain number of years allows students to simulate payment schedules and to determine the "best" loan that they can obtain.

A major advantage to using a spreadsheet to help students to develop algorithms is the speed with which they can start constructing their own sheets. With approximately 12 hours of instruction students are able to construct complex sheets that perform a variety of functions. Additional classroom time permits introduction of tables and table lookup.

On the other hand, several weeks of instruction is necessary before students can write solutions to problems using a programming language. Time must be spent discussing algorithms before language concepts can be covered and then students start the abstract process of coding their solutions. In addition, students are normally taught to design and program the entire problem before attempting to execute it. This method does not permit the immediate feedback obtained when the solution is developed using a spreadsheet.

Except for languages like COBOL, much of the notation used in high-level languages is mathematical and abstract. For example, table lookup that is so easy to use in a spreadsheet is highly difficult to use in a programming language because of the array notation. The matrix notation, which is highly mathematical, is difficult for students who lack a strong mathematical background. Languages, like COBOL, that are much more English-like and more "user friendly" have a tremendous amount of "overhead." Students must learn about compilers, editors, executing programs, and even job control language, before they can execute a program. In either case (mathematical/business languages) attempting to teach students to learn to program solutions to problems with only a few hours of instruction is impossible simply because of all the peculiarities of the programming.

The one area in programming not discussed above is repetition, repeating instructions over and over. This particular topic can be covered using the MACRO capability in spreadsheets. A macro

IS a program; that is, it is a set of instructions that the computer will carry out in a step-by-step fashion given a starting point. However, macro construction is not recommended for an introductory class because it IS programming. Students must be extremely comfortable with spreadsheet design before attempting to use macros.

Programming vs. Databases

Studying packaged database software permits students to further develop their analytical skills without the major handicap of learning a language. Packages, such as PFS File and IBM's Filing Assistant can be learned with only a couple of hours of classroom instruction. Armed with this knowledge, students can construct their own filing systems and develop databases from information supplied by the instructor or from projects they research on their own.

The author often uses a personnel file that would contain such fields as name, address, age, sex, years of work experience, years with the company, and specifially coded work skills. A typical database would contain 25-30 records. Students become familiar with database concepts and techniques for permitting quick entry of data.

In order to obtain information from the database, students learn how to construct queries. This requires learning logic skills and truth tables. For example, one possible request for data would be to obtain a listing of all female personnel, under the age of 30, with previous work experence who have been with the company for less than 5 years. Another example would be to obtain address labels for all employees who have worked with the company for more than 15 years who have a particular coded skill.

Students in programming classes will often build multidimensional arrays to hold information similar to that discussed above. Again, the problems encountered with array notation are numerous. The concept of how the information is stored becomes a major focus, rather than how to obtain information. Instructors will often build data files on disk and let students read the information one record at a time. Thus, students have no real concept of how the data is stored. The IF statement is used to inquire whether the information is appropriate (SEX=FEMALE, AGE<30, etc.).

Using a database removes a major hurdle that students must cross before being able to understand how information can be accessed. The database is the "tool" needed to provide quick and easy access with minimal effort. An analogy might be made with tools used by carpenters. Rather than invent a tool for a particular application, a carpenter will purchase one already made. The emphasis, then, is on the result, and not on the construction of the tool used to get that result.

Conclusion

The argument that every individual "needs to know" something about programming is an archaic one. The software packages available today permit the computer to be used as a valuable tool in problem solving, provided the user knows how to analyze and design a step-by-step solution to the problem. Spreadsheets and databases proliferate the microcomputer software market and although available under a variety of names, they have a remarkable commonality in that all use essentially the same formulas. Therefore, the learning transfer from one package to another is immediate.

Students no longer need the heavy dose of programming theory that may have at one time been essential. For example, in 1930 it was probably necessary that drivers knew as much about their automobiles as mechanics because garages and gas stations were not plentiful. However, because of advanced technology, even mechanics quite often do not know how to repair a part. It is simply an electronic module that is pulled when bad and a new one put in its place. The same concept is true of computer software technology advancements; very complex analysis can be done with a minimal amount of concentration on the "whys" and a concentration on the "hows."

The mathematical background required for spreadsheet and/or database design is not as stringent as it is for programming. Thus, students in grades as low as 7th and 8th would quickly learn to design formulas and to make queries. In particular, those students who are concurrently taking algebra (or just completed it) would be more than prepared to begin to use spreadsheets for practical applications.

In conclusion, spreadsheets and database software provide valuable alternatives to programming to those who have no need to know programming. With minimal effort students can learn to do protocol applications and enjoy the fruits of their labor rather than experience the frustration encountered in so many programming classes.

FIGURE 1

	A	B	C	D	E	F	G	H	I	J
1		Pay		Overtime		Gross		Fed	Total	Net
2	Name	Rate	Hours	Hours	Pay	Salary	FICA	Tax	Deduct	Pay
3	Bill	$10.50	40	______	____	______	____	____	______	____
4	Sue	15.00	40	______	____	______	____	____	______	____
5	Tom	7.20	30	______	____	______	____	____	______	____
6	Alice	8.90	45	______	____	______	____	____	______	____

Contact: M. Gene Bailey
East Tennessee State University
Computer Science Department
P.O. Box 23, 830A
Johnson City, TN 37659

615-929-5332

ADCIS Proceedings, November 1986

The Administrative Evaluation of Microcomputer-Based Instructional Materials Development: A Case Study

George H. Conklin

North Carolina Central University

ABSTRACT

The number of microcomputers found in postsecondary educational institutions has been increasing greatly in recent years. Unfortunately, instructional software in the social sciences in general and in sociology in particular has been and remains scarce. All too often, in many disciplines, emphasis is put on pure research in the university setting. Were a faculty member to develop a software package for instructional use, such a person would run the risk of failing to receive any credit in the academic setting. Development of software, therefore, remains dependent on foundation funding not only for the initial package, but for upgrading as well. This article reports on a university policy where faculty have been recognized for development of software. Upgrading and new versions now figure into the merit pay system, solving the problem of who maintains software already developed. How the policy came about is described, and it is concluded that in the future professional associations must pay attention not only to what is taught, but how and by whom the subject is presented.

WHO EVALUATES ACADEMIC TEACHING AND PUBLISHING?

In most postsecondary educational institutions in the United States, faculty members are expected to contribute to the knowledge-base of their disciplines. Published articles and books are almost always considered to be solid evidence of professional development, provided the books and articles have been refereed by ones peers, and appear with the imprint of acceptable publishers. Journal articles, in particular, provide a relative quick process of evaluation. One or more articles per year in a scholarly journal is not at all uncommon for many professors, not counting presentations at professional societies or other professional obligations.

Even if it is teaching which in fact pays the bills for a university professor, the development and marketing of a textbook may not count as a professional publication for a professor. Rather, textbooks may well be considered teaching materials and be included as a part of the teaching evaluation. The situation is even more complex when it comes to the development of computer-based learning systems. While faculty committees understand the role of textbooks, computer-based materials for microcomputers simply do not fit into the conventional reward system. Because computer-based instructional systems are difficult to develop, and take a great deal of time to perfect, it is only natural that there has been no rush by professors in the United States to develop new sofware. As an example, there are well over 60 different books written for the principles of sociology market, but there are only two or three software packages for the same market.

Although professional articles are evaluated by peer groups, teaching itself is often evaluated by surveys given to students asking questions about the professor. The result is that colleges all too often end up evaluating professional publications through peer review, but may very well evaluate teaching materials through much less formal means. Describing how small colleges operate, an article in THE NEW YORK TIMES WINTER SURVEY ON EDUCATION wrote about small colleges and quoted a professor as saying that '...teachers here 'are committed to teaching, not writing,' ...(a) common characteristic of small colleges. As one indication of the emphasis the school places on teaching, student evaluations of teachers play a significant role in determining promotion and tenure (Perlez 1986:34).'

At first glance it seems to make sense that teaching colleges would promote teachers on teaching as evaluated by students. But let us consider some of the not-so-obvious implications of the quote from Perlez. First, professors at teaching colleges obviously do not have to publish research, since they concentrate on teaching. Perhaps this statement is accepted as normal. But, it also means that such teachers also do not publish teaching materials either. Rather, they teach using textbooks and computer programs written by others, perhaps at research institutions or by independent authors.

As indicated by Perlez above, while journal articles are evaluated by peers, teaching is evaluated by students. But, in the social sciences it is very easy for a professor consistently to make statements in a classroom which would not ever pass a peer review. Would one bother to read a journal for which articles had been refereed by freshman or sophomores? It is even reasonable to ask such a question? Professors who do not publish are therefore evaluated principally by students, and such evaluations may well include such questions as how up-to-date a professor is, even though a student would have no way of knowing.

The above discussion would seem to indicate that at the present time it would seem that many college teachers are not actively publishing research, but they are also not active in developing any type of teaching materials. Rather, the non-research oriented teachers seem to be evaluated by students, whose opinions may or may not be an accurate gauge of quality of teaching or instruction. The results are unfortunate, since the system makes it almost impossible for a professor to develop microcomputer-based software and receive any credit for it in the academic setting. For the purposes of this paper, the logical question to ask is 1. who should develop computer-assisted instructional software, and 2. who should evaluate the quality of such software. It is assumed below that it is most desirable for professors, not computer professionals, to develop software, and that such software must be professionally evaluated. Unfortunately, the current reward structure of most universities encourages neither. We report here on a successful case-study where an administration modified rules for merit rasies and promotion to take into account the development of microcomputer-based software.

WHO SHOULD WRITE AND EVALUATE SOFTWARE?

The thesis of this paper is that professors who teach are those who are best suited to develop software. Is it realistic to expect a professor to write software if it counts toward teaching and is evaluated only through student response?

These two questions were raised by the author with the administration of North Carolina Central University in 1978. North Carolina Central University has a yearly evaluation form for faculty which assigns points ranging from 1 to 5 for both teaching and research. Categories under teaching are clarity, presentation, communication, assignments and materials, and evaluation techniques (exams). Under the non-teaching section (40 percent or less weight) are advising, departmental service, scholarship (including professional growth) and public service. In the above categories, development of software is not mentioned specifically, nor is it easy to say in which category, if any, software development should be placed.

A total of 500 points may be rewarded to each faculty member as a maximum, with a minimum of 100 possible. However, the main issue is who gets recommended for a merit raise. There is no firm cutoff for this. Therefore, in 1980 the Chancellor published a set of criteria he would accept for approval of a merit raise. These included publications, election to a significant office in national or regional professional associations, presentation of papers at meetings, unusual instructional achievement, or unusual community service. Unusual instructional achievement was further defined as 'special teaching techniques or instructional materials which are exportable.' This was interpreted to include software accepted for distribution by a publisher.

The net effect of this ruling was that as a Chairperson of a department I could reward a faculty member with a merit raise for writing a software lesson each year. A merit raise at North Carolina Central University ranges from $500 to $1000 above the normal increment decreed by the state legislature. Thus, a teaching faculty member who developed a software lesson could see a modest salary increase. Several faculty members responded by writing good material, but one stated that $500 (that year's merit amount) would mean that software development was a minimum wage occupation and he declined to participate. For a younger faculty member a $500 raise in base pay would translate into many times that since future increments would include the amount in the base salary.

However, when as Chairman I made the recommendation that a software package accepted by a major publisher be acceptable as basis for a promotion, the issue had to be examined much more carefully. Traditionally good teaching may have been the basis of merit raises, but promotion also required publications. How would software count?

The issue was discussed with the administration using the arguments stated in the first part of the paper. In particular, the author claimed that good teaching required that actual teachers produce their own software, and that a referee's statement that the content of the software was good should be accepted not only a evidence of teaching, but of scholarship as well. While this statement not be true in all disciplines, writing good software in sociology often requires careful research into data bases, research on which variables are linear and non-linear and how to apply such data to concrete situations. Development of computer-assisted software in sociology can be proof that a faculty member has mastered the newer techniques of analysis and has the ability to explain complex data simply.

The argument was not initially successful, since traditionally the university had considered anything to do with teaching materials to be teaching and not scholarship. To get this policy changed, it was necessary to point out that if professors who teach do not write their own textbooks or software, then those with little experience teaching will do the job. The result, it was also argued, was why there has been so little software available for teaching in the social sciences. Although professional programmers like to think that they will do a better job than a teacher, the argument was also made that the dismal quality of much instructional software was that it was designed not by teachers by by computer professionals not attuned to the needs of students.

POLICY CHANGES AND IMPLICATIONS

As a university where teaching is important, North Carolina Central University is faced with a faculty where many have not continued to keep up with their field of study by going to professional meetings or publishing in refereed journals. The next argument used by me as chairperson of a department was that a faculty member who has developed programming skills, written lesson scripts and then had this work refereed by outside sdources has certainly proved that his or her skills are up-to-date. Therefore, a faculty member who lets his or her peers at other institutions read his lessons as they are programmed on the computer has effectively found a way to have those outside the home institution evaluate his or her teaching and mastery of the discipline.

Why teaching should be peer reviewed was the final question which had to be answered. Do students not know good teaching when they see it? Unfortunately, the answer to this question is at best a solid maybe. Even with a discipline such as math where exam results are relatively easy to grade, and where emotional issues such as social policy are not present, students have been shown to be unable to explain accurately how much they have learned. An interesting series of articles in SCIENCE about 10 years ago brought out this issue. Gessner (1973) found that students rank most highly those professors from whom they have learned the most, but Rodin and Rodin (1972) found the exact opposite. Zelby (1974:1268) found no relationship, but the findings suggest that students like professors who show the least intellectual uncertainity. Conklin (1978), writing about social science students, showed that students had strong opinions about which instructor they liked, but that this did not correlate with actual learning. Thus, my final argument ran, having peers outside the institution review teaching was just as important as having outside peer review of journal articles. In both cases, such review would show a faculty member was teaching up-to-date findings in an acceptable manner.

In the fall of 1984 the Dean of the College of Arts and Sciences announced that from that year forward, software packages distributed by professional publishers would acceptable not only fo merit raises, but for consideration for promotion as well. The result of this decision was that it is now possible for me as Chair of a department to promise a faculty member on-going administrative support for microcomputer software development. The development of new software is acceptable for a merit raise. Following acceptance by a publisher with peer review, a professor will be eligible for consideration for a promotion or tenure. Software alone is not sufficient for promotion, but it will count. Time taken developing and maintaining software is now rewarded.

At the 1985 ADCIS meetings presentations by one foundation suggested that one problem for foundation support of software was that the postsecondary software market had failed to develop a way to sustain itself. Due to the labor-intensive nature of the work, support seems necessary not only for development but for revisions as well (Ehrmann 1985:274). In this article I have outlined the arguments used at a university where teaching is important to gain administrative and financial recognition for the development of computer-based instructional software for microcomputers within the traditional academic setting. It is hoped that other professors and administrators will be able to use the ideas presented here to help modify existing academic policies in the United States to encourage development of new computer-assisted instructional programs. However, the future of computer-based instructional materials depends ultimately on how well such undertakings are rewarded not by foundations, but within the traditional academic setting and within academic departments.

REFERENCES

Conklin, G. H. (1978). 'The Frustration of Academic Innovation.' TEACHING SOCIOLOGY 5 (January):125-140.

Ehrmann, S. (1985). 'Looking Up a Funder's Perspective on the Post-Seconary Uses of Technology.' PROCEEDINGS OF THE 27TH ADCIS CONFERENCE:274.

Gessner, P. K. (1973). 'Evaluation of Instruction.' SCIENCE 180 (11 May):566-569.

Perlez, J. (1986) 'The Many Things That Make 'Small' Attractive.' THE NEW YORK TIMES EDUCATION WINTER SURVEY (5 January):34-35.

Rodin, M. and B. Rodin (1972). 'Student Evaluations of Teachers.' SCIENCE 177 (29 September):1164-1166.

Zelby, L. W. (1974). 'Student-Faculty Evaluation.' SCIENCE (29 March):1267-1270.

"A Nationally Validated, Exemplary Computer Management System"

Debra J. Glowinski

E. Wayne Roberson, Ph.D.

Abstract

The CADPP (Computer Assisted Diagnostic Prescriptive Program) is a customized curriculum management software program that is nationally validated as an Exemplary Educational Program. The program is a member of the National Diffusion Network that disseminates effective innovations in education and is sponsored by the U.S. Department of Education.

The CADPP software system is currently used by more than 1,000 school districts in 40 states and includes programs in compensatory education, migrant education, special education, adult education, gifted and talented, junior colleges as well as the regular classroom. The microcomputer curriculum management system manages the necessary paperwork generated by the diagnostic/prescriptive instructional approach in order to provide the teacher more time to teach. The program provides a systematic monitoring of student progress, skill mastery, and teacher use of instructional materials. The program can be used in any curriculum area, provided there is a list of skills and concepts with related instructional materials.

The CADPP software program operates on the Apple IIe, Apple II+, TRS 80 Models III and IV. The program requires 64K two disk drives, an 80 column card, a monitor and printer. The IBM PC version will be available in the Fall of 1986. The software program is revised and updated periodically at no cost to the user and can be copied within the school district for all users. Supportive diagnostic tests in Mathematics (grades 2-6) and Reading (grades 2-9) are available. Only one day's training is required to install the program.

The program disk has the capacity for 150 skills in two curriculum areas. A student data file (100 per teacher) includes name, grade level, age, gender, reading level, and learning modality. A prescription file for 1,600 instructional activities per disk is also included. The files provide information to generate the following reports: an individual student prescription, with six activities per skill, a skill mastery report, and a summary of prescriptions completed by each student.

"A NATIONALLY VALIDATED,EXEMPLARY COMPUTER MANAGEMENT SYSTEM"

The CADPP (Computer Assisted Diagnostic Prescriptive Program) is a computer managed software program which was validated in reading and mathematics by the National Institute of Education and the United States Department of Education in the fall of 1979. Since that time, the CADPP has been funded by the National Diffusion Network to provide inservice training and materials to educational service providers in replicating the program. The CADPP computer management system has been disseminated nationally and adopted by over 1,000 school systems in 40 states.

The CADPP is a curriculum management system which allows the user to load: 1) learning characteristics of individual students; 2) a skills/concepts list; and 3) skill oriented characteristics of available instructional materials into a computerized file/report system. Once the files are loaded, the CADPP software program cross-references the files to match similar student characteristics to related instructional materials, this correlation generates a customized prescription (personalized educational plan) for each participating student, based upon the skills/concepts requested by the teacher.

Instructors utilizing the diagnostic/prescriptive instructional approach can benefit by having the microcomputer manage the paper work necessary in the implementation and monitoring of individualized instruction, thereby leaving them more time to teach.

ABCDEFGHIJKLMN

The CADPP management system generates individual student prescriptions based on skill needs, provdes a systematic monitoring of student progress, skill/concept mastery, and teacher instruction. The software program was developed by ECIA Chapter 1 teachers inorder to facilitate their efforts in meeting the diagnostic/prescriptive instructional approach to teaching reading and mathematics as required by the federal regulations.

DIAGNOSTIC/PRESCRIPTIVE

READING
PHONICS
RELATED SKILLS

Although the CADPP computer management system was developed to assist ECIA Chapter 1 teachers in the implementation of mathematics and reading programs in grades 2 through 9, it has the flexibility to be adopted to any curriculum area at any grade level. Only two things are required: 1) a list of skills and concepts to be taught, and ; 2) related instructional materials. Currently the software program is being used in Adult Basic Education Programs and at the junior college level in the Basic Skills Competency Programs. In addition the CADPP program is widely used in Special Education Programs, Migrant Education Programs, Gifted and Talented Programs and Compensatory Education Programs. This program is not for every teacher, but will be of interest to those teachers who tend to use a diagnostic/prescriptive approach to instruction. In a sense the computer becomes a teacher aide, managing the paperwork, reporting and monitoring of instructional activities.

Evaluation results from pre/post achievement test scores in mathematics computational skills and reading comprehension consistently indicated greater gains as compared to program results prior to the installation of the CADPP computer management program. Continued monitoring of the program indicates that the gains are sustained from year to year. In addition, the program once installed provides a basic structure for the novice teacher to follow. To assist teachers in the diagnosis of basis skills needs, a battery of formative and summative criterion referenced tests(CRTs) are available for the 121 mathematics skills (grades 2-6) and the 77 reading skills (grades 2-9). The mathematics skills include the basic numeration and computational skills taught in grades 2-6. The reading skills includes phonics (grades 2-3), related skills (grades 2-6) and comprehension skills (grades 2-9). The summative CRTs' have established reliabilities and standardized administration and scoring procedures.

The original CADPP software program was developed to operate on an IBM mainframe computer, within a year it was modified to run on the NCR and CADDO minicomputers. In 1980, it was written to run on the Apple II+ microcomputer. The current CADPP software operates on the Apple II+, the Apple IIe, the Radio Shack TRS 80 Models III and IV. The IBM PC version will be available in the Fall of 1986. Hardware requirements include, two disk drives, 64K, an 80 column card, a printer and a monitor.

The CADPP software program disk includes two files. A skill/concepts file that provides for a maximum of 150 skills in two curriculum areas. The program disk comes with 121 basic mathematics skills and 77 basic reading skills on file, but each skill can be deleted and replaced with other skills or concepts. Up to 100 students per teacher can be placed on the student file. Student information required by the file includes name, gender, teacher, birthdate, grade level, class, learning modality, and instructional level. In order to create the activities file, a disk is initialized and formated to include data for 1,400 instructional activities. Information entered for the instructional activities include skill number, teacher number, activity name, modality, instructional level, interest level, equipment required, time required. All files can be modified, expanded or deleted.

EDUCATIONAL MATERIALS

After the files are loaded, the user can then request individual student prescriptions by simply entering a student number, and the number of each skill for which a series of six prescriptions will be printed. The program is limited to a maximum of six skills per student per request. In addition to individual student prescriptions, other reports can be generated. A progress report lists the students, prescriptions completed and skills mastered. The skill mastery report is a summary of all skills mastered by a student to date.

The CADPP computer management system includes a program disk and a user's manual. One day of training is required to install the program. Educators adoptin the program, may copy the software for use within their school district or institution. Technical assistance is available by telephone at no additional charge. Periodic revisions and updates are made and the new version of the program is available to users at no charge other than a floppy disk and postage.

COMPUTER LITERACY
MYTHS AND DELUSIONS

Tommas J. Koehler

Montgomery County Public Schools, Rockville, MD

ABSTRACT: This paper reviews the rationale behind the computer literacy "movement" and offers a critical view of the stated goals of computer literacy courses. Course content is also critically reviewed.

Old and new arguments for computer literacy are reviewed. Computer literacy is seen as a mythological need and is often presented by the ignorant using archaic concepts. Futurists initially believed that, with the proliferation of microcomputers, knowledge of computers and how to use them would become a fundamental skill. Technical knowledge and skills relating to computers (or Computer Literacy) were considered basic skills and these skills were thought to be as important as writing, reading, and mathematics. Failure of schools to make a major commitment in the area of computer literacy was seen to have disastrous consequences for both the educators of the public and the future of education.

However, the successful and widespread proliferation of microcomputers in a non-computer literate populace seems to weaken any arguments for universal computer literacy. The need for general technical knowledge and skills in using computers is highly questionable and the need for general technical computer literacy is declining rather than rising. The tendency to increase the simplicty and naturalness of computer use has long been promised and is now being realized with the drive toward simplicity having been adopted by the microcomputer industry with a vengeance.

The information presented in K-12 computer literacy and programming courses often does not meet minimal levels of professional practice, nor is it taught by persons with adequate training. What we are seeing in secondary school computer literacy and programming is a reflection of obsolete ideas and academic unreality in colleges.

The author concludes by saying that some teaching and learning about computing would be valuable, but only to a specific student population.

In the early 1980's computer literacy was a new growth industry in education. Various "experts" appeared in board rooms and staff conferences to raise consciousness and stir the field into action, while the educational literature carried article after article proclaiming the good news.

The thesis of this concept was that due to the proliferation of small, inexpensive and all pervasive microcomputers, everyone would need to understand and be familiar with these devices. Knowledge of computers and how to use them was predicted as a fundamental skill for the future. In addition to the proliferation of microcomputers, an additional drive was the picture painted by futurists who saw a shift from the industrial to the information age. Information as a profitable resource and product would require skilled workers to produce it. (1) As shall be shown, these literacy skills were interpreted to be mainly technical skills including programming computers in relatively low-level languages.

This original plea for computer literacy was well summed up by J. E. R. Licklider of MIT, "Our education is not only missing an opportunity by not using computer technology more widely, it is failing to discharge a responsibility. Whether or not schools will be a serious factor in educating a computer literate public by 1990 is an open question. The failure of schools to make a major commitment in this area now can have disastrous consequences for both the educators of the public and the future of public education." (2)

This bandwagon of the future was answered in the plans and budgets of many school systems. In announcing the 1983 budget for the nation's 11th largest school system, Fairfax (VA) County's then Superintendent, Dr. Linton Deck, stated, "There's no way we can expect our students to function in the 21st century without training in the basics of dealing with computers." (3) This statement was made in justification for a one-million-dollar budget for classroom computers for computer literacy. Another superintendent of a major school district supported a similar budget request as follows: "I believe that computer literacy will be as basic as reading and writing in the years ahead." (4)

These statements clearly indicate that many top school administrators were convinced that knowledge and skills relating to computers were *basic* skills, and that these skills are as important as writing, reading, and mathematics. Statements like these were very typical of the feelings of many school administrators and appear to have been widely accepted by school boards and the public with little questioning.

One representative example of an attempt to define Computer Literacy is from the project conducted by the Human Resources Research Organization (HumRRO) of Washington, D.C. This project, supported by the National Science Foundation, was intended to develop "universal" computer literacy by infusing computer related skills and knowledge throughout the K-8 curriculum.

HumRRO's definition of computer literacy ran as follows: "knowledge of computer applications, understanding of the impact of computers on society, skills in using computers, and some knowledge of how computers work." (5) Some specific objectives of this computer literacy curriculum are more specific:

Develop procedures to organize data
Read program documentation
Retrieve information from a computer database
Use a word processor
Code a program
Test and debug a program
Recognize main components of computer systems. (6)

These concepts were echoed at the 1983 National Educational Computing Conference in the various presentations in the Computer Literacy "track." These presentations, as in the HUMRRO example, indicate a high *technical content*--one presentation going so far as to present a simplified *machine language* simulation as "an important subject for computer literacy." Thus a high proportion of computer literacy curricula was developed with a detailed *technical* component which emphasized programming in languages such as LOGO, or BASIC. (7)

These definitions and objectives, therefore, constitute a good picture of the "Basic Skills" which were seen to be required in the future by all functioning members of society. Note that what was being said is that *all* students should be able to do technical things such as *code, test and debug a program*, and that these are basic skills.

The passage of time and accumulated experience suggest that American education should not be restructured to teach "computer literacy as a fourth "R". Developments indicate that the fundamental logic of the arguments about the need for "computer literacy" is false. There can be no argument that microprocessors are everywhere--in cars, microwave ovens, games, ad infinitum, and that small computers have many uses (good, bad, and indifferent) as educational and business tools. However, the computer literacy issue has no particular relationship to these facts. Indeed, the successful and widespread proliferation of microcomputers in a "noncomputer literate" population seems to weaken any arguments for universal computer "literacy."

The fundamental fallacy of the computer literacy idea relates to the conclusion that sometime in the future *all* functioning members of society will need *technical* skills to *use* these small computers. In actual experience, it is clear that the need for *general* technical knowledge and skills in using computers is highly questionable and that, indeed, the need for general technical computer literacy is *declining* rather than rising.

There is no doubt that microprocessors (computers) are appearing in all sorts of strange and wonderful places, and that information processing systems and terminals are now proliferating into the smallest business operations. Computer controlled manufacturing and programmable robots are on the rise. However, the *general* user of these devices has need for *far less* technical computer knowledge than in the past. The "transparency" of computer operations to the user is increasing, and thus the knowledge and technical skills needed are decreasing.

The essential simplification of computer usage for the general user is well summed up in Daniel D. McCracken's fine article on "Software in the 80's." (8) Mr. McCracken states that the software of the 80's will be "User Friendly ... a new buzzword signifying Simplicity of Learning," which will "... make simple jobs really simple." Much of the activity such as simple report generation now done by technicians will be done by end-users (i.e., nonliterate, nontechnical) in increasing numbers. (9)

This tendency to increase the simplicity and "naturalness" of computer use has long been promised, and is now being realized. The CRT terminal has led to self-educating systems, and voice response (already in use) and artificial intelligence concepts are removing the need for the general user to learn the minor intricacies of the computer. Many current and successful future computer programs will be so natural that they will almost eliminate the issue of training. (10)

This drive toward simplicity has been also adopted by the microcomputer industry with a vengeance. Apple Computer's Trip Hawkins states that "ease of use will be the most critical factor in future (software) designs. You have to take some functions, make them rich but always aim at making them easy to use. For the novice, you'll have to hide powerful features behind simple designs." The vast bulk of the use of microprocessors is one where they are preprogrammed. The operations performed are not intended to be changed, and the only way the device can be "fixed" is to replace a component.

General literacy (knowledge and skills) relating to computers, and process control microprocessors in car ignitions, TV games, microwave ovens, and vendor software etc., will do no one any good, as it cannot be exercised without violating a warranty or disassembling the unit.

In this light, perhaps one of the most insightful criticisms of computer literacy as a technology oriented course is by Bill Wilkerson, a noted writer and expert on ATARI Systems. Wilkerson states:

> When computers are as much a part of everyday life in this country as automobiles are now (and I firmly believe that they will be), then I think they will be treated much as automobiles are.
>
> In California, the State has decreed that all high school students shall take a course in "computer literacy." So what happens? Every high school is scrambling to buy one or two computers and begin teaching every kid how to program in BASIC. Great, right? Nonsense!
>
> I think it is senseless to equate 'computer literacy' with 'learning to program in BASIC.' After all, 'automobile literacy' consists of learning traffic laws, safe driving techniques, and actually starting to drive a car (it's usually called 'Driver Training').
>
> 'Automobile expertise,' on the other hand, consists of learning what tools do what, the theory and practice of internal combustion engines, and how to maintain and repair an automobile (and this is usually called 'Auto Shop'). Does every student take driver training? Yes, or nearly so. Does every student take auto shop? No. Not by a long shot.
>
> So, I believe, it should be with computer literacy. Don't teach everyone how to program. (What would we do with a nation of programmers? The same thing we would do with a nation of auto mechanics?) Instead, teach everyone how to use a computer to do word processing, to balance their budget, to access data bases, and the list could be quite long.
>
> And, yes, keep the computer programming classes. But keep them on the same basis that auto shop classes are offered--as electives, for those interested in learning more than how to 'drive' their computers or cars.
>
> Why this confusion of computer literacy and computer expertise among schools and teacher? Partly because the computer industry has promoted the view.
> (Perhaps fearing that current applications programs are inadequate to a classroom situation?) Partly because of a dismal lack of education and information on the part of the educators. Pity the poor math or history teacher who is nearing retirement. Suddenly he/she is forced to learn enough about these nasty machines to be able to teach some kids how to use it. Do you wonder that the path of least resistance is most often chosen? (11)

Dr. Joseph Weizenbaum, a computer science professor at MIT, summed up the issue well: "Just as the encyclopedia salesman was able to convince parents that their kids would fall behind the other kids if they didn't buy an encyclopedia, so the computing industry, including all the computer side-products such as magazines, have invented the problem of computer illiteracy to increase their sales." (12)

Another argument relates to the use of microcomputers and other computer hardware in the "world of work," and in our daily lives. All business and administrative areas increasingly rely on computers to function. The truly modern supermarket with no marked prices and scanners for checkout cannot function without the computer. However, the *general* computer knowledge required to operate in such an environment is now less than it would have been in the past. Other than some trivial knowledge mainly for reassurance and a list of absolute "don'ts" the supermarket personnel need to know nothing. The main knowledge required is what *not* to do. The National Commission for Employment Policy at a recent conference recognized this by labeling the computer literacy idea as "obsolete" and ill-advised. (13)

Up until 3 to 4 years ago it was to endanger one's life to criticize computer literacy courses. Times have changed and the critics are now emerging in increasing numbers. Many writers now focus on the educational intent--the "fit" in the curriculum of standalone computer literacy courses. They do so quite correctly. However, if one believes that the monster, once created, will live on, we had also better concentrate on what goes on in these courses. A cynical view would say that since these computer literacy courses are not going to disappear, they should be examined and their grosser errors cleared up.

When one examines the literature and practices concerning the teaching of items relating to computer "literacy," one finds that the content of these courses is below any reasonable professional standard, is often obsolete, is frequently useless, and sometimes erroneous.

The information presented in K-12 computer literacy and programming courses often does not meet minimal levels of professional practice, nor is it taught by persons with adequate training. It is an interesting exercise to chart a few of these egregious errors based on information presented in the literature--presumably as "good practice."

Structured Programming, Analysis and Design

The last major article on flow charting in the professional computer literature was probably published sometime before our current students were born, but flow charting as a tool in 9-12 computer programming courses lives on. (14) Flow charting is a common component of computer literacy courses. Flow charting is an obsolete and archaic method of graphically depicting computer systems. Flow charting was developed for "Unit Record" (punch card) equipment in the 1940's. It is no longer used in state-of-the-art software development. It has long been superseded by other methodologies. Teaching flow charting is analogous to teaching drafting using T-squares instead of drafting machines and CAD/CAM Systems.

Instead of teaching flow charting, thereby giving it a cachet of learnedness, the techniques of structured programming design and analysis including their *graphic* methods should be taught. Since there are several of these structured technique with individualized concepts and graphic conventions, decisions must be made as to the most appropriate to teach. The two leading candidates are the Warnier/Orr an Yourdon/Constantine techniques. (15) Both Yourdon and Orr have commercial firms which popularize these techniques, and could most likely be persuaded to work with educators in developing an introduction to these techniques. At the very least, simple methods such as HIPO Charts should be incorporated into the course content.

The Teaching of Programming Rather than Systems

Most computer literacy courses seem to be compelled to have an objective requiring the student to write a "program." Programming courses, of course, are just that. In the real world environment programs are very late in the life cycle of *systems* and are mere building blocks in greater wholes. This concept is destroyed by the drive to have students write meretricious "programs" to do standalone trivial functions. Programming, *per se*, is a pernicious skill, which professional data processors largely eschew. Systems thinking, top down design, modular programming, and systems architecture and structure are ignored when standalone "programming" is taught. (16)

Teaching Esperanto

The teaching of computer languages so that students can actually "do something" has surface educational merit, and has been a hallmark of most computer literacy and programming courses. The teaching of BASIC has been profoundly damned by the proper gurus, but continues unabated; repeating the condemnation is unnecessary. The rush to teach PASCAL--in order to satisfy Advanced Placement Test pipedreams--is the newest rage. PASCAL deserves the same level of condemnation as BASIC, but for different reasons. PASCAL has severe limitations as other than a professor's toy, has not made major inroads as a commercial language, and is, in fact, outdated. (17)

The key concept that has escaped computer educators is that any computer language is relatively good for some defined purpose-- none are universal. PASCAL is very good for taking the advanced placement test.

Why not teach COBOL--it's useful, marketable, has loads of good text books, confronts the student with solving structural problems and will never go away. It is not sexy, exotic, does not confront students immediately with hard core computer science concepts, nor will it help pass AP exams. *However, 80% of all The world's current computer programs are written in COBOL*. English is the most *taught* language in the world. Esperanto is a better structured language, but it isn't useful. Application of this same logic should apply in the computer field. (18)

One tongue-in-cheek suggestion would be to convert either BASIC or PASCAL to run using Latin commands. The student could learn two dead languages at once--at least one of which would be useful.

However, given the microcomputer orientation of computer literacy/programming education in the K-12 curriculum and the desire to meet a very vague group of college computing literacy expectations, at the least switch should be made to "C," the microcomputer language of the future. It has as much, or more, learned cachet as PASCAL, is growing into the base language for micros, is transportable, and has as many or more computer science concepts imbedded in it as PASCAL. (19)

The teaching of BASIC and particularly PASCAL is directly analogous to basing the automotive mechanics' curriculum on Wankel engine work and body repair of Rolls Royce's. Technically interesting to the connoisseur but neither useful nor marketable, and often above the level needed by the student.

Literacy for the College Bound Computer Student

One of the *only* cogent reasons for computer literacy and programming classes is to prepare the college bound student who intends to enter a computer oriented program. Secondary school computer literacy and programming courses tend to be reflections of the perceived desires of college professors. Quite frankly, nothing could be worse. The thrust of most college computer speciality degree programs is generally considered by practitioners to be unrealistic and completely "out of touch with revolutionary changes in software technology and business applications." Secondary school computer literacy and programming course content generally reflects both the ivory tower and archaic "old think." James Martin, undeniably one of the most influential innovators in the computer software field, lambastes this tendency succinctly. Martin categorically and vehemently indicts the universities:

> Neither the business schools nor the computer science schools have gotten anywhere near understanding what's happening, let alone teaching it....I think many of the top people in computing at the present time are extremely dismayed by what's happening in the computer science schools. I was at an important facility of one of the world's largest computer manufacturers recently and the general manager of that facility commented that he would, literally, never hire a computer science graduate. That's an indication of the extent to which the computer science departments are perceived to be out of date.
>
> Martin contends that universities have largely ignored the technologies of the 80's and are continuing to teach the obsolete methodologies and languages of the 70's. (20)

What we are seeing in secondary school computer literacy and programming is a reflection of obsolete ideas and academic unreality in colleges. At best a computer literacy/programming set of courses prepares students to learn more (in college) of something that is, charitably, a dinosaur.

Education by the Uneducated

Little firm research data is available on educating the computer educator. This is probably wise because if it were exposed and compared to other academic disciplines, it most likely would appear scandalous.

The English, science, or mathematics teacher is rigorously trained in a body of knowledge and in how to teach it. The computer "whatsit" teacher is usually created by a process more related to religious conversion than academic training. A summer workshop and an in-service course are the equivalent of an undergraduate degree, and a college course or two added on is better than a doctorate. Intensity, concentration, and rigor is not present, neither is much content. Not only is the exposure short, but weak and watered down.

Dave Morsund, in an excellent editorial, "More Harm than Good?" quotes a colleague who discovered a not atypical effort to create computer literacy "teachers." The two-week course was to convert physical education teachers into computer literacy teachers. Morsund paraphrases his colleagues' outrage--"Would you like your children's first serious computer course to be provided by '14-day wonders' from a random academic field? How can computer educators allow such a thing to happen? It's a crime. Something should be done." (21) Unfortunately "something" is seldom done, and this is not an overly extreme example of creating teachers of a complex subject by methods more ceremonial than real.

Mass Marketing and Education

Some socio-economic speculation might hold that computer literacy and programming are taught because of large advertising budgets combined with boredom. One day's viewing of television will convince (and has) the general public that lack of computer education will doom their children to flunking out of college, or not being able to manage their cake factory. The Charlie Chaplin clone and his like have created the demand. Bored teachers and closet computer cadets, who often are willing to do anything to avoid grading more papers on geometry theorems, have supplied the educational part of the product--computer literacy courses.

None of this is to say that some teaching and learning about computing would not be valuable to a specific student population. It does remain for the appropriate computer literacy concept to be formulated, focused, debugged, and put in place with trained teachers to teach it to those who need it. Instead of participating in an orgy of "hula hoop educational knee-jerk curriculum development," it might be wise to learn from the mistakes of the last few years, back up, and really figure out the what, why, and how about this issue. The only other approaches are to let it roll on, or to whimsically hope that it will go away when replaced by better things.

Computer Literacy as a general subject is a mythological need, and is often presented by the ignorant using archaic concepts. Given that general computer literacy should not be in the curriculum, perhaps the fact that its content is poor should be shrugged off. However, if there is indeed a "crisis" in education, perhaps the millions (perhaps billions) being poured down this drain would be better invested. Policymakers should consider the judgement of history, technology and the classroom. David Tyack and Elizabeth Hansot sum up this idea: "The past suggests that we should be cautious about predicting a technological revolution in the schools. One hopes that the computer will not become just another piece of machinery gathering dust in the corner. But it can never replace the interactions between teachers and students that, at their best, produce literacy and civility."

NOTES

(1) "Interviews with John Naisbitt," Computer Decision, Sept. 15, 1983, pp.174-176.

(2) National Science Foundation, Technology in Science Education, July 1979, p.1.

(3) Washington Post, 6 January 1982, p. C7.

(4) Gaithersburg Gazette, 7 January 1982, p.21.

(5) HumRRO, My Students Use Computers. A Guide for Computer Literacy, Summary, August 1981.

(6) Ibid., P. III-4.

(7) Proceedings of NECC/5, June 1983, pp. 180-181.

(8) Daniel D. McCracken, "Software in the 80's," Computerworld, 17 September 1980, pp. 5-10.

(9) Software News, 4 January 1982, p. 9.

(10) David Meyers, "Bricklin Says Software Still has a Way to Go," Computerworld, Dec. 12, 1983. p.13.

(11) Bill Wilkerson, "Insight," Compute, December 1983, pp. 264-66.

(12) Joseph Weizenbaum, quoted in "MIT Professor Pans Use of Computers in Schools," Deborah Anker, Spring 1986 Class Bulletin, Montgomery College, Rockville MD, p. 38.

(13) Education Computer News, Vol. 3, No. 12, June 4, 1986, pp. 1-2.

(14) Tina T. Cheng, "A Validation Study of the Computer Literacy Examination: Cognitive Aspect," AEDS Journal, Spring 1985, Vol. 18, No. 3, pp. 148-149. (The CLECA attempts to measure the concepts that should be taught as computer literacy. Two items of this thirty-two item test deal with flow charting.)

(15) Victor Weinburg, *Structured Analysis*, Yourdon Press, New York, 1980. (This book is essentially a survey of structured techniques. It is an excellent introduction, but needs to be supplemented in the detailed use of any specific technique.)

(16) Edward Yourdon and Larry Constantine, *Structured Design*, Yourdon Press, New York, pp. 27-57.

(17) John Dickinson, "Programming Makes Sense for Business," *PC*, Vol. 4, Number 22, October 29, 1985, p. 114. (This issue of *PC* specialized in reviewing microcomputer languages and is an excellent survey of all major microcomputer languages.)

(18) Ibid; pp. 125-131

(19) Ibid; pp. 153-161

(20) "An Information Systems Manifesto," *Communications of the ACM*, March 1985.

(21) Dave Morsund, "More Harm than Good" (Editorial), *The Computing Teacher*, December/January 1984-5, p. 3.

ADCIS Proceedings, November 1986

COMPUTER ASSISTED COURSE DEVELOPMENT AND INSTRUCTIONAL SYSTEM (CACDIS) *

A.L.Lakshminarasimhan
Associate Professor
Computer Science Dept.
Stevens Institute of Technology
Hoboken, N.J., 07030

ABSTRACT

This paper describes the design, development and implementation of a 'Computer Assisted Course Development and Instructional System' (CACDIS) on Digital PC 350. This system can be used by instructors to create Computer Aided Instructional (CAI) materials for a variety of courses. Students can access the CAI materials through this system for self study. Also the instructors can use the system as an aid in class room teaching. This system also provides an easy interface with graphics to create pictures. These pictures can be combined with textual informaton. The use and the capabilities of the system are illustrated with an example of a graphics course. The proposed system would enhance human-machine interaction both as instructional aid to students and also for the instructors for the generation of the course material.

INTRODUCTION:

In a major study on computers and their use in the educational environment, a carnegie commission noted that there are two problems of computers as instructional tools. Firstly, there is a lack of good learning materials in many areas. How do we generate such learning materials? Secondly, the non-availability of learning materials, to students on a large scale basis. With the advent of personal computers there has been a remarkable growth in their use for instruction and education for solving the above problems and as a consequence we are no longer limited by the availability of resources of a large time-sharing system. The use of microcomputers is a powerful medium for instruction. Thus the hardware needs of a typical computer-based tutorial or instruction is not excessive and is well within reach of students.

Historially, Computer Aided Instruction (CAI) has been used by industries to train their personnel on specific aspects of a job. In CAI the student trains himself by repeatedly going through a sequence of instructional material. Persons with varied learning abilities could progress at their own pace as they receive the feedback from the CAI. This process is difficult and almost impossible if an instructor were required to pay individual attention to each one of the students. Now CAI is being increasingly used in classroom education at schools and universities. The use of visual aids is certainly an asset, especially where there is a need to portray graphics information. Needless to say, even simple animation can be a very effective means of understanding complex ideas in Science and Engineering. The use of CAI with animation makes the process of instruction very effective. However, the creation of the course material for CAI is rather tedious and time consuming. This paper describes a Computer Assisted Course Development and Instructional System (CACDIS). It is simple to use and suitable for course material development and to impart instruction.

The CACDIS system provides the user with two functions - the creation of a new course and accessing of an existing one. Creation of a new course involves three subfunctions; create the necessary menu using Frame Development Tool (FDT), create textual information and create graphical information. The

---

*Also presented at 1986 Spring DECUS U.S. Symposium menu frame enables easy interaction for students to traverse through different parts of the course material. This is created by using the FDT available on Digital PC 350. The instructor interactively creates pages of tutorial material by properly assigning chapter numbers, section numbers and page numbers. The system interactively reads this information and stores it as a record (with concatenation of chapter number, section number and page number as an index to that record). The facility includes provisions for easy modification and deletion of pages. One could also insert a page between two existing pages. During the creation of course material, the instructor can traverse through the course material so far developed by using 'NEXTSCREEN' and 'PREVSCREEN' keys. Also, the course material could be in any arbitrary order. The graphics generator of CACDIS enables creation of graphical (pictorial) information. The instructor can use the core graphics language primitives interactively to draw pictures.

Students can access the relevant course material through CACDIS as follows. First the CACDIS displays the information regarding the chapters of the course selected. The student/user can then make his selection. The system then displays the first page in a chosen chapter/section and the student can traverse through the course forwards or backwards by using NEXTSCREEN and PREVSCREEN keys. This feature of traversing through course material is also present during creation and modification of course material.

The CACDIS provides 'help' feature. Both instructors and students can query for operational details while they are either creating course material or accessing it.

The main features of the CACDIS system are the following:

1) Students constantly interact with the course material while learning.
2) A great deal of individualization of the course material is possible to suit the needs of the students.
3) An aid to students in problem solving skills through computer aided instruction.

PROBLEM DEFINITION

We are concerned with the design and development of a CACDIS system having the following capabilities.

a) An efficient tool for instructors to create course material consisting of textual information and graphical information.

b) A facility for comfortable learning of course material.

THE CACDIS SYSTEM

The CACDIS system is intended for two types of users. The 'instructor-user'- the person who intends to use the system and generates a computer based course material. The 'student-user'- who uses the system and learns a particular course.

The system design depicting various modules and their relationships is illustrated using the hierarchical diagram of fig. 1. The CACDIS is divided into two modules:

1) Module to create a new course.
2) Module to access an existing course.

The creation module is divided into three sub modules:

1) Course structure
2) Manu and help frames through FDT
3) Textual/graphical information

The course material is input in terms of pages. The system is menu-driven with the 'help' feature at each stage of development of course material or retrieval of course material.

FILE SYSTEM

The CACDIS system requires creation, modification and access of pages containing CAI material. The file structure chosen should be capable of accessing the pages randomly as well as sequentially. Hence we have chosen index sequential access method provided by RMS on the Digital PC 350. In order to accommodate the facility of inserting pages between existing pages, the primary key of index sequential file is chosen as chapter number, section number, page number and additional page number.

PROGRAMMING DETAILS

All the modules, illustrated in Fig 1, are coded in PASCAL. In what follows, we describe the scenario of operation of the system.

The 'Instructor-user' creates the course by invoking the CACDIS system. A main menu, as shown in fig. 2, will be displayed on the screen. He may choose to create a new course by positioning cursor at 'Creating a new course' and pressing Do. Then the menu, as shown in fig. 3, will be displayed. He would create course structure which is planned earlier for computerising the instruction.

A. Creation of Textual Information

Textual information is stored in blocks of 1280 characters called 'pages'. Once we select option 'Create textual information' from the menu (see fig. 3), then a new menu as shown in fig. 4, pops up. This has four options which are described below:

(i) Create a new page

The system prompts for the chapter number, section number, and page number. If the page does not exist, then the system displays a blank page. The user may enter the textual information. The process is terminated by pressing the 'Do key'. However if the page is already created then a prompt to that effect is given to the user. An example for creating page 1 of chapter 1 section 1 is shown in fig. 5.

(ii) Page Modification

Once the user selects this option, the system prompts for the chapter, section and the page number. If the page exists then it is displayed as shown in fig. 6. If the user intends to modify this page, he could respond 'yes'. System then positions the cursor at the beginning of the page for modification. If the user responds with a 'n' then the modification is not done.

(iii) Page Insertion

An additional facility of inserting a page before an existing one can be done in two ways:

1) Two pages in the same chapter and section, with numbers say 1 and 2 are created by the user. If the user later needs to add another page between these two pages, then he can respond with a 'y'.

If the user's response is 'n', then the page traversing menu is displayed and the user may traverse the database. After each page is displayed the system enquires if the user intends to insert a page before the current one.

2) Suppose there are two pages existing in chapter three, section four-(pages five and ten with additional page number of zero). The page being currently diplayed is page ten. The user prompts the system that he intends to insert a page before the current one. The system then allows the user to create a new page with identification of chapter three, section three, page number nine and additional page number zero.

(iv) Page Deletion

Existing pages in the data base may be deleted using this facility. The system prompts the user for the chapter number, section number and page number. The data base is checked for the existence of this page and it is displayed (see fig. 8). The system issues prompt to confirm deletion. The user may then delete this page by responding 'y'. If however, the user decides not to delete the page, the page traversing menu is displayed. For each page that is displayed the system prompts the user to see if he intends to delete the current page.

B. Creation of Graphical Information

This module provides the user with a facility for generation of picture and text for illustration. A large number of instructions chosen from core graphics language is implemented.

As in the case of text creation, the system prompts the user for the chapter, section number and page number. The system checks for the existence of the page and then prompts the user to that effect if the page does exist. If it does not exist then the screen is cleared and core graphics primitives are displayed. Additional core graphics language instruction can be seen by depressing 'NEXT SCREEN'. Further, the user can traverse back by pressing 'PREVSCREEN key' for earlier core graphics language instruction. The user enters the number corresponding to the instruction that is needed and the system prompts with the corresponding inputs needed. After the user gives inputs for the instruction, an effect of execution of instruction is shown interactively on the screen.

C. Accessing the Course

Now the CACDIS system is ready for accessing course material. The student-user can go through series of chapter, section numbers and pages to learn about a specific topic (see fig.9).

CONCLUSION

CACDIS is a general purpose system with dual capability of creating course material for a specific subject and its use as Computer-Assisted instructional system for students. The CACDIS system, implemented on Digital PC 350, is powerful enough to be used for the development of computer based instructional system. It is simple and is completely menu driven. The users of CACDIS can learn the operation of system easily with the 'help' feature provided at each stage of operation. It is important that the present day computer aided instructional packages provide an easy interface for creation of pictures. The course material created for students may not only contain textual information but also pictorial drawings. Accordingly, CACDIS is provided with an easy inteface to graphics. This does not need elaborate programming usually required and hence it is easy for user to create graphical images. Such a CAI material on CACDIS would significantly enhance understanding of course.

REFERENCES:
1. A.L.Lakshminarasimhan "Computer Assisted Course Development and Instructional System on Digital PC 350", Computer Science Dept. Report, Stevens Institute of Technology, 1986.

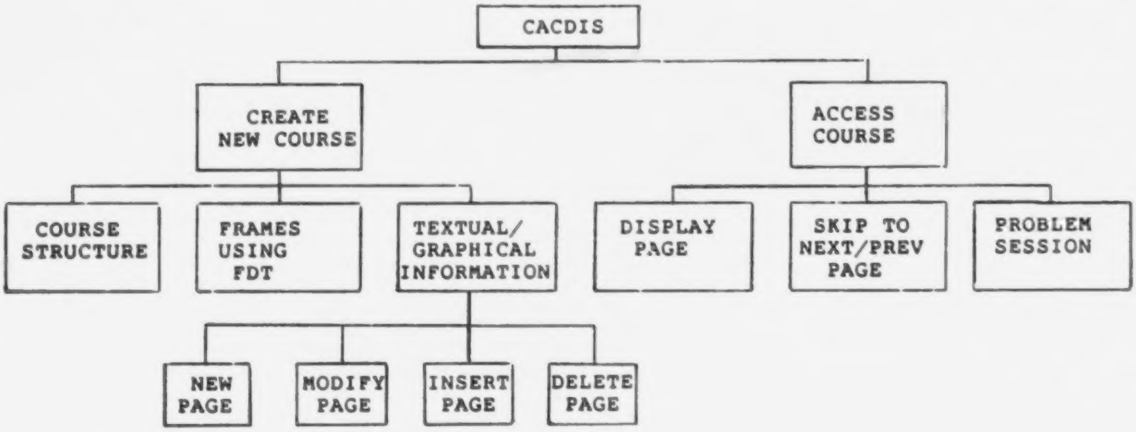
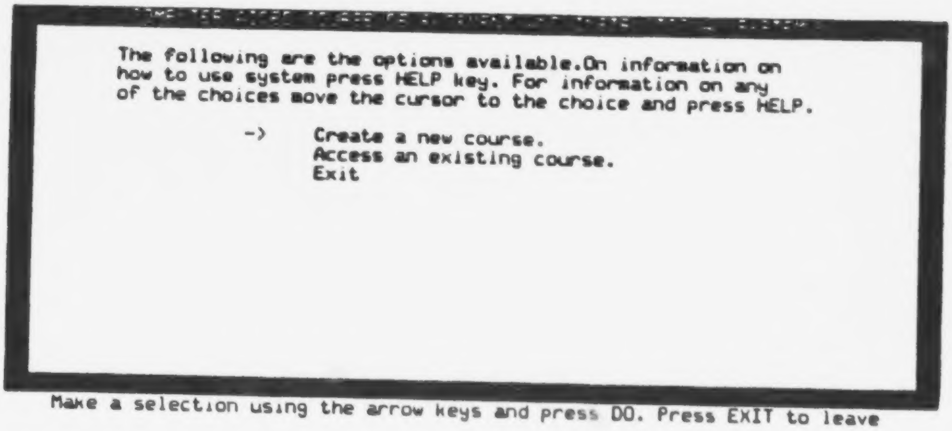
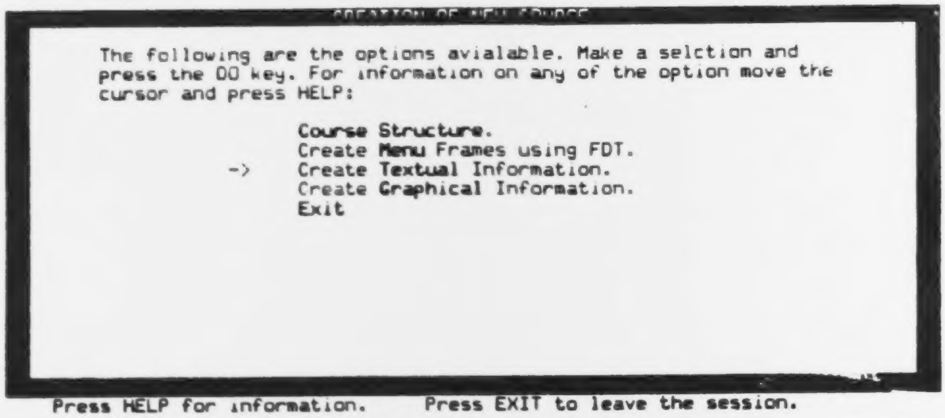

CACDIS
CREATE NEW COURSE
ACCESS COURSE
COURSE STRUCTURE
FRAMES USING FDT
TEXTUAL/ GRAPHICAL INFORMATION
DISPLAY PAGE
SKIP TO NEXT/PREV PAGE
PROBLEM SESSION
NEW PAGE
MODIFY PAGE
INSERT PAGE
DELETE PAGE

Fig 1: Hierarchical diagram of CACDIS system



```
[illegible]

The following are the options available.On information on
how to use system press HELP key. For information on any
of the choices move the cursor to the choice and press HELP.

        ->     Create a new course.
               Access an existing course.
               Exit

Make a selection using the arrow keys and press DO. Press EXIT to leave
```

Fig 2



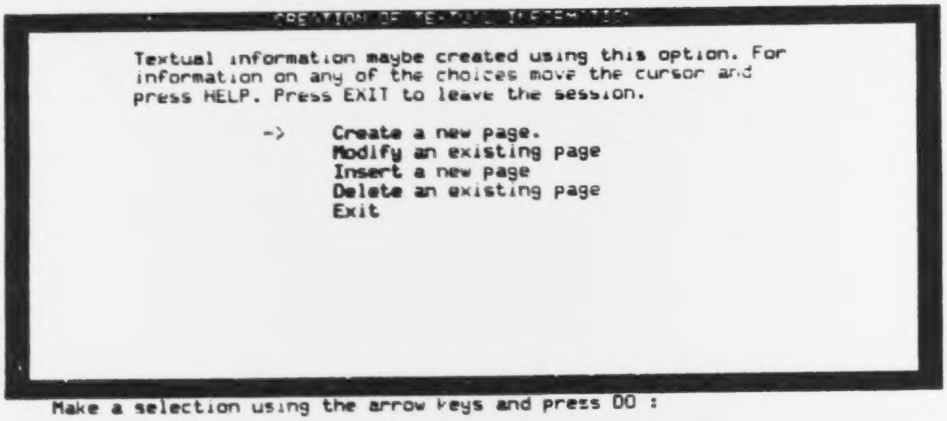
CREATION OF NEW COURSE

The following are the options avialable. Make a selction and press the DO key. For information on any of the option move the cursor and press HELP:

Course Structure.
Create Menu Frames using FDT.
-> Create Textual Information.
Create Graphical Information.
Exit

Press HELP for information. Press EXIT to leave the session.

Fig 3



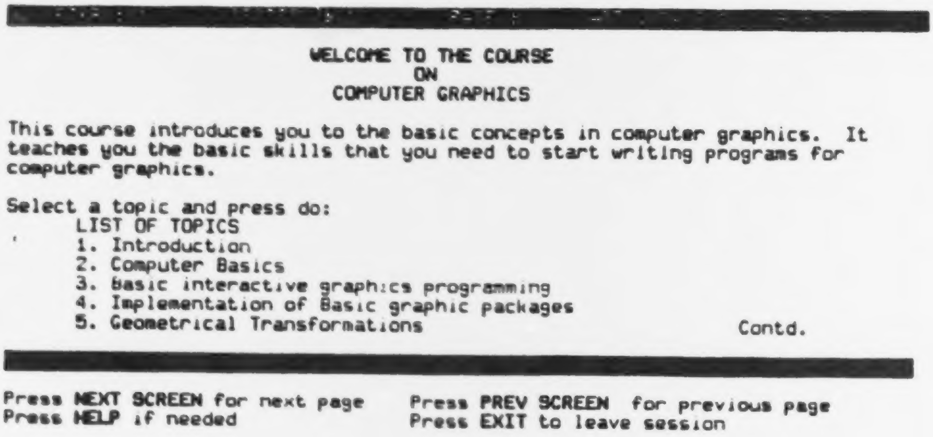
CREATION OF TEXTUAL INFORMATION

Textual information maybe created using this option. For information on any of the choices move the cursor and press HELP. Press EXIT to leave the session.

-> Create a new page.
Modify an existing page
Insert a new page
Delete an existing page
Exit

Make a selection using the arrow keys and press DO :

Fig 4



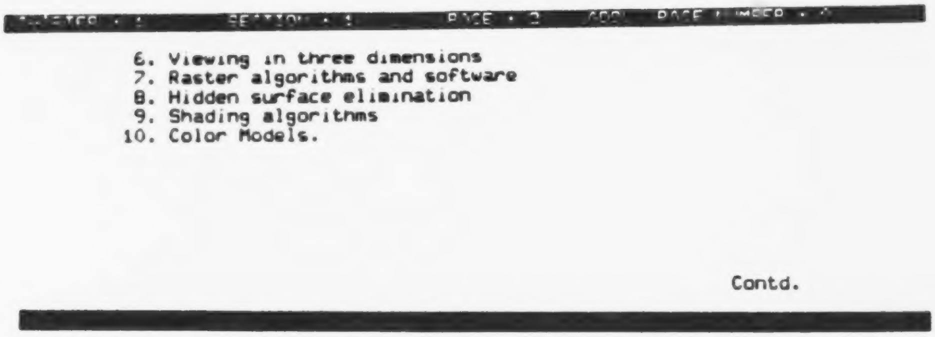
WELCOME TO THE COURSE
ON
COMPUTER GRAPHICS

This course introduces you to the basic concepts in computer graphics. It teaches you the basic skills that you need to start writing programs for computer graphics.

Select a topic and press do:
LIST OF TOPICS
1. Introduction
2. Computer Basics
3. Basic interactive graphics programming
4. Implementation of Basic graphic packages
5. Geometrical Transformations
Contd.

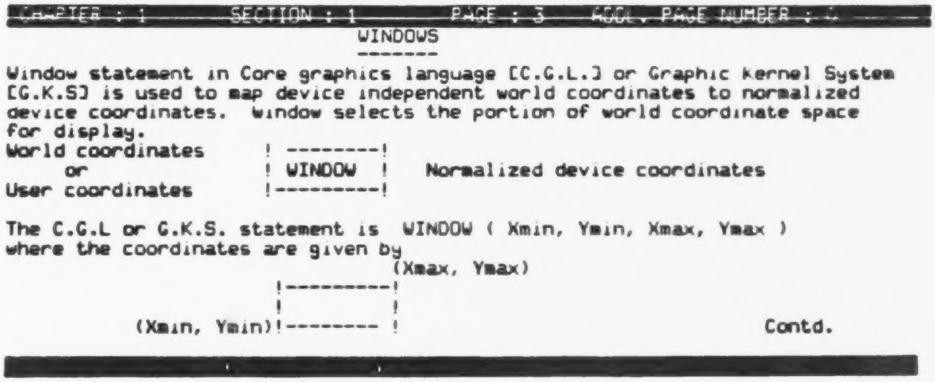
Press NEXT SCREEN for next page
Press HELP if needed
Press PREV SCREEN for previous page
Press EXIT to leave session

Fig 5



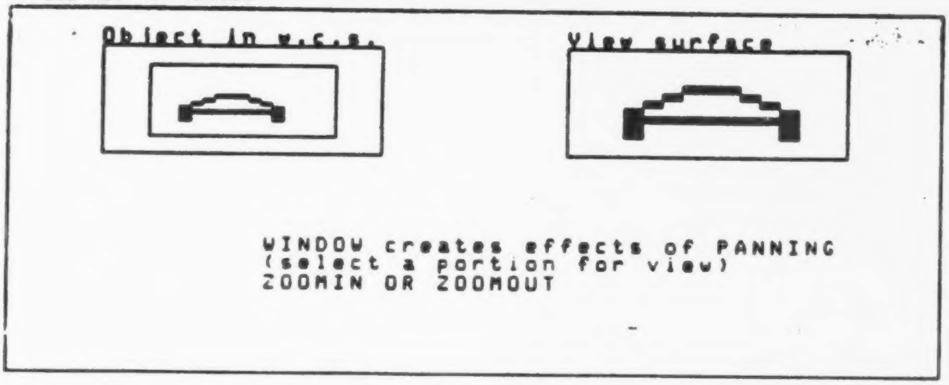
```
CHAPTER : 1      SECTION : 1      PAGE : 2    ACCL. PAGE NUMBER : [illegible]
           6. Viewing in three dimensions
           7. Raster algorithms and software
           8. Hidden surface elimination
           9. Shading algorithms
          10. Color Models.

                                                          Contd.
```

Fig 6 MODIFY THIS PAGE ? [Y/N/HELP]:
Press HELP if needed



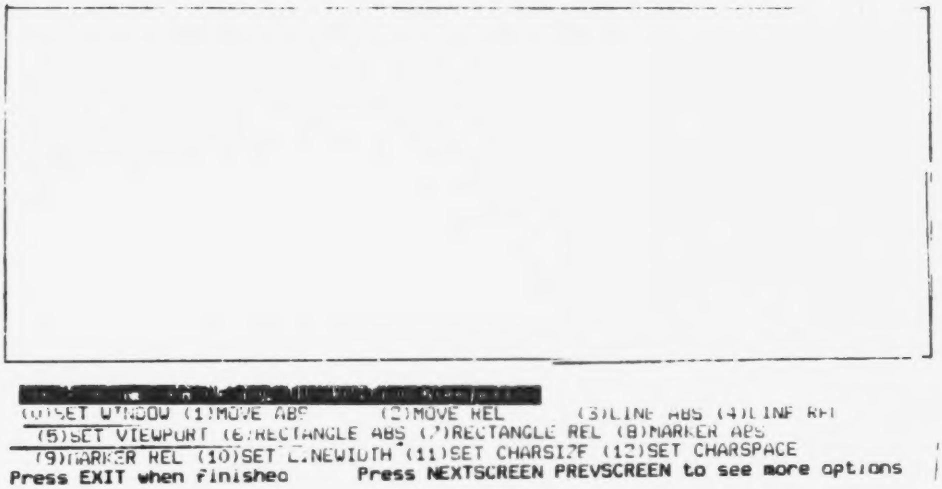
```
CHAPTER : 1      SECTION : 1      PAGE : 3    ACCL. PAGE NUMBER : [illegible]
                                 WINDOWS
                                 -------
Window statement in Core graphics language [C.G.L.] or Graphic kernel System
[G.K.S] is used to map device independent world coordinates to normalized
device coordinates.  window selects the portion of world coordinate space
for display.
World coordinates      ! --------!
     or                ! WINDOW  !   Normalized device coordinates
User coordinates       !---------!

The C.G.L or G.K.S. statement is  WINDOW ( Xmin, Ymin, Xmax, Ymax )
where the coordinates are given by
                                  (Xmax, Ymax)
                       !----------!
                       !          !
          (Xmin, Ymin)!--------- !                              Contd.
```

Fig 7 INSERT BEFORE THIS PAGE ? [Y/N/HELP]:
Press HELP if needed



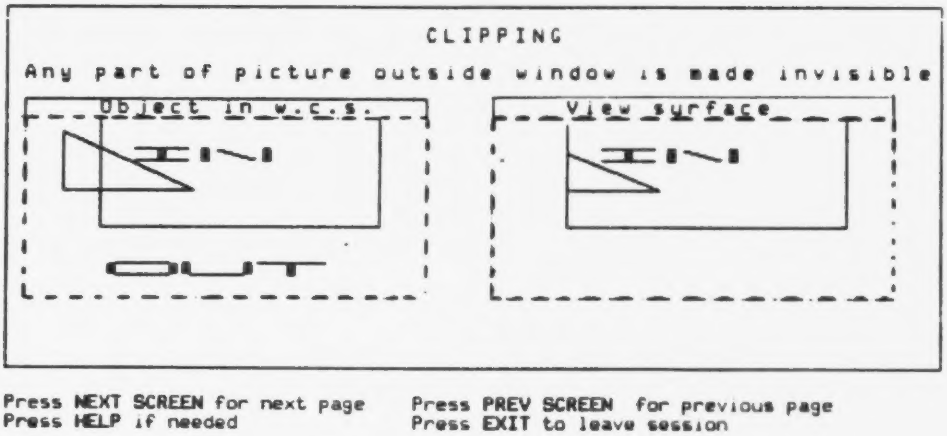
```
WINDOW creates effects of PANNING
(select a portion for view)
ZOOMIN OR ZOOMOUT
```

Fig 8 DELETE THIS PAGE ? [Y/N/HELP]:
Press HELP if needed



```
[illegible]
(0)SET WINDOW (1)MOVE ABS      (2)MOVE REL      (3)LINE ABS (4)LINE REL
 (5)SET VIEWPORT (6)RECTANGLE ABS (7)RECTANGLE REL (8)MARKER ABS
 (9)MARKER REL (10)SET LINEWIDTH (11)SET CHARSIZE (12)SET CHARSPACE
Press EXIT when finished      Press NEXTSCREEN PREVSCREEN to see more options
```

Fig 9



CLIPPING

Any part of picture outside window is made invisible

```
Press NEXT SCREEN for next page      Press PREV SCREEN  for previous page
Press HELP if needed                 Press EXIT to leave session
```

Fig 10

STREAMLINING THE SOFTWARE DEVELOPMENT PROCESS THROUGH DESIGN AND CODING GUIDELINES

Mary Jac Reed
University of Delaware
Office of Instructional Technology

ABSTRACT

The University of Delaware has recently completed the development of a large courseware package. The package consists of thirty individual modules comprising a one-semester, college-level tutorial and testing course. The key to keeping the development schedule was the time devoted at the beginning of the project to develop coding and design conventions that were firmly adhered to by all designers and programmers on the project.

INTRODUCTION

Software development is a time-consuming proposition. Many methods have been devised to shorten the time committment--authoring systems, unimaginative drill and practice applications, team approaches, to name a few. It is difficult to produced quality software within a reasonable timeframe. Written guidelines for both the design and the coding of software is also a time-consuming task but one that benefits not only the present project but future development efforts as well.

PLANNING

At the University of Delaware, a rigorous team approach is employed in the software development effort. Subject matter experts are university faculty members and the instructional designers and programmers are professional persons from the Office of Computer-Based Instruction (OCBI). The development of a project timeline is the first order of business when a new project begins.

After twelve years of software development projects, OCBI has good estimates of the time it takes for the various stages in the development process. A brief outline of the process follows. The faculty members have prepared an outline of the idea for the computer when they write a formal proposal to OCBI requesting support. If their proposal is funded for development, then several hours per week of an instructional designer are assigned to the project. The designer's main function is to adapt the faculty person's content outline to the computer media in the best possible way, both technically and educationally. After rounds of review and revision, with input from other content experts, other design persons, and programmers, the material is approved by the team members and enters the coding phase. The programming design is usually planned by a full-time programmer while the actual coding of individual lessons is entered by student programmers. Every part of the project team, designer, programmer, and faculty member, reviews the lesson during the coding phases, suggesting rewording and revisions as the lesson develops. After the lesson is approved by the teams, it is piloted by a small number of students drawn from the target population. The student pilot results in many more revisions and rewordings. A full-scale student use is planned after this revision stage.

With large courseware packages that contain many modules, it is advantageous for the entire team to provide consistency from module to module. However, with most large development efforts, there are usually many people making decisions at the module level, such as the designer, faculty person, coder, programmer, and reviewer. Individual designers and programmers tend to introduce more, not less, inconsistency. A project manager, overseeing that written standards are, first of all, reasonable and, secondly, adhered to by all team members, is the essential element in a successful team approach to software development. Consistency is a quality that requires much communications between various parts of the development team. The benefits are seen by the team in a more smoothly flowing development process and benefits are seen by the end user in a more reliable and predictable software product.

If the communications is done very formally at the onset of the project, in the form of specific, written guidelines, much time is saved in the production of the final product.

Written guidelines help considerably in the production of large courseware packages. But large courses are not the only instance when guidelines are beneficial. Carefully considered standards once written can be applied to future development efforts in other content areas and improve the quality and efficiency of all future production.

DESIGN GUIDELINES

What are design guidelines? The guidelines can take many forms--questions that the developer needs to ask AND answer, a template that the developer "fills in" with content, an authoring system used for development, a formal document stating the design standards of a project.

In the courseware package that I will use as my example, the designers, faculty, and programmers created a template--an empty shell of a module--that contained the design of various displays. The more attention that is paid to detail at the beginning planning stages, the more time is saved and the more consistency is built into the modules. A warning to the novice developer, however, is that experience is the best aid in making the correct decision. Therefore, the first effort towards writing guidelines will be a good step in the right direction, but these initial guidelines will probably be revised at the beginning of a new project.

Examples of the kind of detailed decisions that were made in the design template are the following:

- What the title page of each module would contain
- If modules would contain objectives and prerequisites displays
- If a display showing available function keys was to be included in the modules
- What person the computer text would use, when speaking to the users (first person, WE, or second person, YOU?)
- How to properly display and punctuate bulleted lists
- What colors to use for various functions
- The organization of the modules into sections and subsections
- How to consistently indicate the first display of a section and the final display of a section
- The wording of directions, question feedbacks, and other messages to users

Selected displays from the design template are included at the end of the discussion.

CODING STANDARDS

Projects usually have more than one programmer working on various parts of the software product, so coding standards are more familiar to most programmers working in structured programming environments than design guidelines would be. Coding standards are probably not adhered to with any more regularity however, due to human nature and individual styles of coding.

Programming languages now are designed to take advantage of multiple programmers, with a modular approach and independently compilable code. With a programming staff of more than two, coding standards are essential.

The coding guidelines that were written for this courseware development effort included the following:

- Modular and unit names follow a certain format, for example, m1s2d3 for module 1, section 2, display 3
- All common routines, such as the index and the branching module, were located in a central library of routines, named in a consistent

manner

All commands and reserved words in the language were UPPER case in the code; all routines from the library were designated with a "ST_" prefix.

At the beginning of each module and each unit, a standard format for documenting the code was included and completed by each programmer, documenting each change and each call to subroutines and variables

Sample coding standards are included at the end of this discussion.

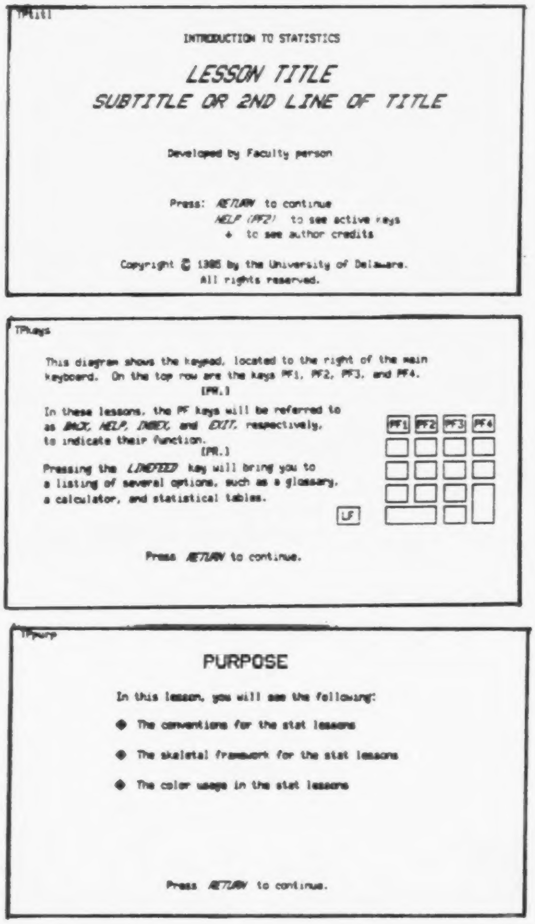
The project manager and coordinator of the student programmers at first needed to coerce and demand that coding standards were met. Code was rejected if it did not follow the prescribed guidelines. As the project grew and more and more programmers were added, the enforcement of coding guidelines grew easier, though, not more difficult. The standards made sense and made coding easier. The new programmers were made familiar with the libraries and with the standards, and that became the way of life on the project.

Programming talent was still needed. The coding standards did not solve all the problems that a programmer was asked to depict on the screen. Rather the standards did away with the more mundane aspects of CBI programming, and allowed the programmers to get into more challenging programming tasks. However, the more complicated tasks were not duplicated. Once completed, these routines also were added to the centralized library and made available to the entire programming team. By the time the project was at its end, and it was time to more on to a new development project, a vast library of routines were ready for incorporation into the next project's modules.

CONCLUSIONS

As a result of writing design and coding standards, the present courseware package was able to be designed by several designers yet retain a similar look overall from module to module. The enormous turnover rate of student programmers could be handled since the modules were well-documented and had a minimum of quirky code from individual programmers. Routines were centrally located and were referenced in the code by name as to which routine was called. Much code could be transferred from one lesson to another as one module was completed.

An advantage that was not anticipated is the shortened timeline for future development. All the design and coding guidelines are in place and have been revised thoroughly during the development of the first package. It would be extremely easy to begin another project in a brand new content area, now that the shells of the displays and the code have been established in the previous development project.



TPtitl

INTRODUCTION TO STATISTICS

LESSON TITLE
SUBTITLE OR 2ND LINE OF TITLE

Developed by Faculty person

Press: *RETURN* to continue
HELP (PF2) to see active keys
◆ to see author credits

Copyright © 1985 by the University of Delaware.
All rights reserved.

TPkeys

This diagram shows the keypad, located to the right of the main keyboard. On the top row are the keys PF1, PF2, PF3, and PF4.
[PR.]
In these lessons, the PF keys will be referred to as *BACK, HELP, INDEX,* and *EXIT,* respectively, to indicate their function.
[PR.]
Pressing the *LINEFEED* key will bring you to a listing of several options, such as a glossary, a calculator, and statistical tables.

Press *RETURN* to continue.

TPpurp

PURPOSE

In this lesson, you will see the following:

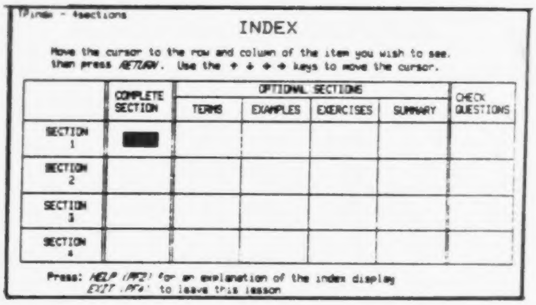
◆ The conventions for the stat lessons

◆ The skeletal framework for the stat lessons

◆ The color usage in the stat lessons

Press *RETURN* to continue.

Fig. 1 Selected screen displays from the Design Guidelines Template



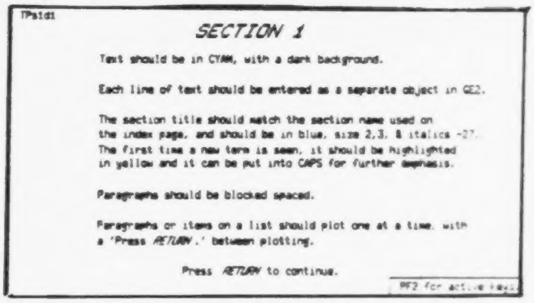
TPindx - 4sections

INDEX

Move the cursor to the row and column of the item you wish to see.
then press *RETURN*. Use the ↑ ↓ ← → keys to move the cursor.

	COMPLETE SECTION	OPTIONAL SECTIONS				CHECK QUESTIONS
		TERMS	EXAMPLES	EXERCISES	SUMMARY	
SECTION 1						
SECTION 2						
SECTION 3						
SECTION 4						

Press: *HELP (PF2)* for an explanation of the index display
EXIT (PF4) to leave this lesson



TPs1d1

SECTION 1

Text should be in CYAN, with a dark background.

Each line of text should be entered as a separate object in GE2.

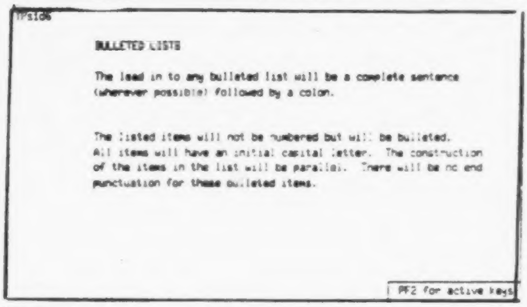
The section title should match the section name used on the index page, and should be in blue, size 2,3, & italics -27. The first time a new term is seen, it should be highlighted in yellow and it can be put into CAPS for further emphasis.

Paragraphs should be blocked spaced.

Paragraphs or items on a list should plot one at a time, with a 'Press *RETURN*.' between plotting.

Press *RETURN* to continue.

PF2 for active keys



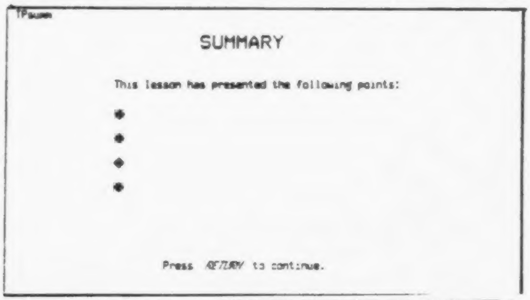
TPs1d6

BULLETED LISTS

The lead in to any bulleted list will be a complete sentence (wherever possible) followed by a colon.

The listed items will not be numbered but will be bulleted. All items will have an initial capital letter. The construction of the items in the list will be parallel. There will be no end punctuation for these bulleted items.

PF2 for active keys



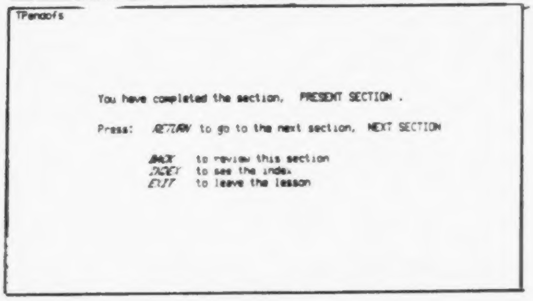
TPsumm

SUMMARY

This lesson has presented the following points:

-
-
-
-

Press *RETURN* to continue.

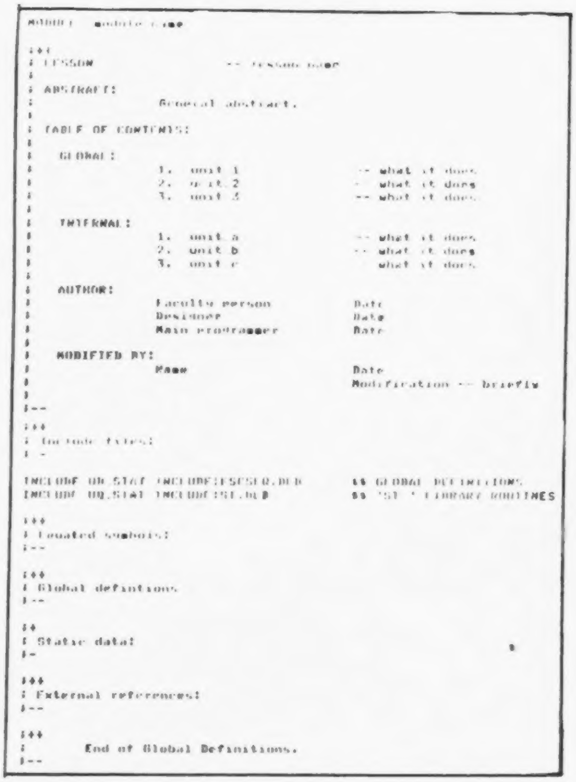


TPendofs

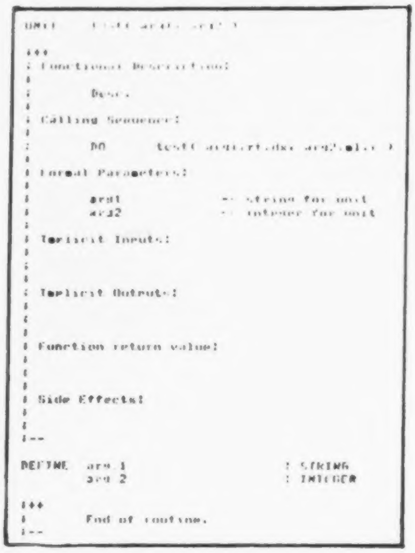
You have completed the section, PRESENT SECTION .

Press: *RETURN* to go to the next section, NEXT SECTION

BACK to review this section
INDEX to see the index
EXIT to leave the lesson



```
MODULE   module name

!++
! LESSON                       -- lesson name
!
! ABSTRACT:
!                General abstract.
!
! TABLE OF CONTENTS:
!
!   GLOBAL:
!                1.  unit 1                   -- what it does
!                2.  unit 2                   -- what it does
!                3.  unit 3                   -- what it does
!
!   INTERNAL:
!                1.  unit a                   -- what it does
!                2.  unit b                   -- what it does
!                3.  unit c                      what it does
!
!   AUTHOR:
!                Faculty person               Date
!                Designer                     Date
!                Main programmer              Date
!
!   MODIFIED BY:
!                Name                         Date
!                                             Modification -- briefly
!
!--

!++
! Include files:
! -

INCLUDE UU.STAT INCLUDE[LESSON].DEF          !! GLOBAL DEFINITIONS
INCLUDE UU.STAT INCLUDE[SL].DEF              !! 'SL' LIBRARY ROUTINES

!++
! Equated symbols:
!--

!++
! Global defintions.
!--

!+
! Static data:
!-

!++
! External references:
!--

!++
!        End of Global Definitions.
!--
```



```
UNIT     test( arg1, arg2 )

!++
! Functional Description:
!
!        Desc.
!
! Calling Sequence:
!
!        DO        test( arg1:string, arg2:integer )
!
! Formal Parameters:
!
!        arg1                 -- string for unit
!        arg2                 -- integer for unit
!
! Implicit Inputs:
!
!
!
! Implicit Outputs:
!
!
!
! Function return value:
!
!
!
! Side Effects:
!
!
!--

DEFINE  arg 1                           ! STRING
        arg 2                           ! INTEGER

!++
!        End of routine.
!--
```

Fig. 2 Documentation to be included at the beginning of each new module and every new unit

ADCIS Proceedings, November 1986

Distributed UNIX-based CBT Authoring Networks

Charles V. Welsh and Thomas T. Chen

Global Information Systems Technology, Inc.

Abstract

This paper presents a rationale for use of UNIX in distributed networks of CBT authoring and delivery systems. Comparisons are made of distributed system features with 1) those found in multiple-user, time shared, mainframe systems and 2) personal computer-based systems. Required network support features are also listed.

Introduction

Existing CBT authoring and delivery products have traditionally tended to center around the two ends of the system configuration spectrum: multiple-user, typically time sharing main frame systems; and personal computer (PC) based systems. Each has a set of advantages associated with it. At the center of the spectrum, UNIX-based distributed networks allow the advantages of both multi-user and PC based systems to be integrated. A rationale will be given suggesting the UNIX operating system in such distributed networks.

Multiple-user System Advantages

Multiple-user CBT systems allow centralized databases of applications programs, courseware, and student data. Additionally, these systems typically feature a large base of indigenous system applications and utilities. Other typical positive attributes also include: ability to service large numbers of terminals, shared access to resources with security, archiving and software maintenance services, access to wide-area data networks, and computation requiring special storage and machine power (Parr, et al., 1985). Inter/intra-system communications capabilities are also important assets as they facilitate the formation of a user community.

PC based System Advantages

Personal computers, without the resource sharing characterized by multi-user systems, guarantee a predictable speed of program execution. Initial startup costs are frequently lower, particularly in smaller configurations, than multi-user systems. PC's also can be easily transported to locations where training is desired.

Distributed System Features

Consisting of a network file system which connects to two or more computers, a distributed system can allow:

1. centralized databases of applications programs, courseware, and student data;
2. processing at workstations within the network, providing a guaranteed speed of execution;
3. communication between network nodes;
4. support of large numbers of terminals, as processing is distributed to subordinate systems rather than supported centrally;
5. flexibility and expandability of configuration - additional nodes may be added as needed. Systems may be distributed in full, in clusters, and hierarchically with predefined superior/subordinate relationships among network nodes.

Required Network Support Features

In this paper, a network is considered to be a group of loosely connected units or nodes. Network nodes may be identified according to their depository (information center) functions, processing functions, and relationships to other nodes within the network (Chen et al., 1976).

A network can be a customized, highly specialized product or it may be a standard product (ie. Ethernet). The exact type of network is not significant, as long as it provides basic support features. These basic support features include:

1. logical distribution of information across the network, independent of the bandwidth and type of the network used. (The bandwidth of the selected network, of course, has implicit performance ramifications);
2. access to required files at whatever node those files may reside on the network, within the context of the preestablished relationships between nodes. (ie. if a user's access to a node is appropriate and/or permissible);
3. real-time communication features among network users;
4. structures to facilitate downstream data transmissions (lessons, assignments, information, etc.) and upstream collection (student data); and
5. network administration facilities including site management capabilities and user access control.

The Rationale for UNIX: UNIX Advantages

As UNIX is inherently a multi-user, multi-tasking operating system, little or no modification of the actual operating system is required for its use in a CBT network. A multiple-user system is highly desirable when high bandwidth communication through the shared memory region is required among various users. This requirement is necessary to satisfy CAI applications such as multiple-user interaction in a team based simulation program. Multiple tasking is important when concurrent tasking is necessary. In the network environment, network interaction can best be handled by a task which is separated from the main CAI courseware task. When done in this manner, the CAI author does not have to worry about the mechanics of network transmission and receiving. In addition, many CAI courseware applications can better be realized by concurrent tasking for a single user.

Examples of the utility of this concept include: windows into several screens constituting a single display, and embedded concurrent training applications in which the training courseware may reside in a concurrent window or may monitor the keys sent to the application being trained.

One of the most critical aspects of using UNIX as a network operating system is the presence of standard network file system facilities. These facilities combine disks residing at various nodes into one logical file system throughout the entire network. The network file system, in turn, allows users the use of distributed processing units without worry of physical disk locations. Two well known network file systems (NFS, proposed by Sun Micro Systems, and RFS, by AT&T) are both UNIX-based. A secondary but also important feature of UNIX is its portability. Because UNIX is written mostly in the "C" high level programming language, it can and has been ported to many computers, ranging from PC's to large main frame systems. This means that initial startup costs can, to a degree, be controlled by the user. Thus, a UNIX network may consist of various types of computers and greatly reduces the risk of required rehosting due to hardware upgrades or expansion. Furthermore, standard UNIX system utilities provide critical functions required for implementation of a distributed network, particularly inter-system communication facilities, necessary for distribution of courses and collection of student data. The layered, hierarchical structure of the UNIX operating and file systems imparts the ability to use an open architecture in the design of CBT authoring and delivery products as well as to customize the user interface. Cost incentives are also present in that additional nodes may be added to a properly designed UNIX-based network as they are required, rather than necessitating tremendous allocation of assets in order to purchase large initial computing and storage resources.

UNIX Disadvantages, Addressed

Historically, there have been several significant drawbacks to discourage use of the UNIX operating system in distributed systems. These obstacles are: the large size of the UNIX system software, UNIX's large memory and processor requirements, and user interface issues.

The UNIX operating system is large, requiring approximately 2 to 10 megabytes of disk space and .5 megabytes of RAM to store and run a reasonable implementation of the operating system and utilities. This does not include resources required for additional applications programs or in which to develop courseware. To accomodate courseware development activities, a more utilitarian computer might contain 30 megabytes of disk storage and 2 megabytes of RAM. While at one time these requirements posed serious problems, the declining cost of memory and mass storage has reduced the significance of this issue; Hirschbuhl (1980) observed storage capabilities increasing by a factor of 32 every decade. Similarly, computational power required by UNIX, once a significant problem in implementing the operating system on smaller computers, has been addressed through common usage of 32 bit microprocessors.

UNIX has frequently been called "unfriendly" by many users who have felt frustration when first interacting with the operating system. Because of its depth, flexibility, frequently cryptic command language, and plethora of jargon, extending into even the most elementary user manuals, many users feel rather overwhelmed when first interacting with UNIX. Research on higher level user interfaces for authors and students has been recognized as a need (Kearsley & Hillelsohn, 1982). While this remains a significant issue within UNIX-based products this problem is beginning to be addressed by software vendors whose products constitute additional layers interposed between the user and UNIX. These products generally intend to provide ways for users to accomplish courseware development and delivery functions whthout knowledge of the operating system command language.

Conclusions

UNIX is viable for use in distributed networks because it:

1. can effectively merge positive attributes of traditional multi-user and PC based CBT systems by facilitating an effective network mechanism;
2. is multi-user and multi-tasking;
3. is achieving industry standardization with availability on all sizes of computers from a variety of manufacturers;
4. provides more hardware and software vendor independence;
5. will become easier to use as more higher level user interfaces are developed;
6. will tend to cost less as storage capabilities continue to increase; and
7. can be used with an assortment of network hardware.

References

Chen T. T. A Depository Health Computer Network, Advanced Medical Systems -- the Third Century. Society of Advanced Medical Systems. 1976.

Hirschbuhl J.J. Hardware Considerations of Computer-based Education in the 1980's. Journal of Research and Development in Education, 1980, 3.

Kearsley, G.P. & Hillelsohn, M. J. Human Factors in CBT. Journal of Computer-based Instruction, 1982 8(4), 74-84.

Parr, F. N., Auerbach, J. S. & Goldstein, B. C. Distributed Processing Involving Personal Computers and Mainframe Hosts. IEEE Journal on Selected Areas in Communications, 1985, 3 479-489.

A REGIONAL STANDARD FOR DEVELOPING AND DOCUMENTING SPREADSHEET TEMPLATES WITHIN THE COOPERATIVE EXTENSION SERVICE

Margaret P. Ezell

Cooperative Extension Service [1]

ABSTRACT

The use of spreadsheets among professional staff within the Cooperative Extension Service has skyrocketed. In a time of tight budgets and scarce resources, templates can, with imagination, be designed as subject matter training tools and simulations. Approximately 80 percent of Extension computer program needs can be addressed by using spreadsheets with templates.

The sections of the paper discuss: 1) the advantages and disadvantages of using templates; 2) problems which arise in using templates; 3) guidelines for designing, programming and documenting easy-to-use templates; 4) procedures for auditing and reviewing templates; and 5) suggestions for training others to use templates with clientele.

INTRODUCTION

"The Computer," states Alvin Toffler, "with its unprecedented power for analysis and for dissemination of extremely varied data, in unbelievable quantities and at mind-staggering speeds, has been a major force behind the acceleration of knowledge acquisition." The computer is forcing changes in the way educators work - changes in the ways clientele are served by the Cooperative Extension Service (an educational unit of the U.S. Department of Agriculture and State Land Grant Universities). The challenge is for educators to anticipate rather than react to the needs of clientele, or some other group will.

Computers are supposed to miraculously improve educators' work with clientele. But the computer will not work without software which is often too expensive for tight budgets, unavailable, or does not work on a particular computer. One method of circumventing this problem is by using what is generically called a spreadsheet program (Supercalc, Multiplan, Lotus 1-2-3).

A spreadsheet is composed of [cells] organized in [rows] and [columns] that are identified by letters and/or digits. There can be two items in a cell - the [contents] (which are typically visible) and a [formula] (typically not visible) for determining the contents. The contents of a cell may be (1) a [value], such as 350.00; (2) a [label], such as MONTH; (3) a special indicator, such as NA for "not available;" or (4) nothing at all, said to be an [empty] cell. (See Table 1.)

	A	B	C	D
1	MONTH	HOUSING	FOOD	AUTO
2	JAN	350.00	200.00	175.00
3	FEB	350.00	240.00	185.00
4	MAR	350.00	250.00	160.00

Table 1. Example of a spreadsheet.

Templates are dataless spreadsheet program overlay files that contain all the calculation formulas necessary to meet specific application needs such as cash flow statements, expense sheets or loan amortizations when the data is added. The power of spreadsheet programs lies in the ability of the spreadsheet to maintain formula relationships without data. Spreadsheets with macro facilities can be written to look like a program in C, Pascal, or Basic. New programs coming on the market such as BALER allow templates to be compiled and run without the spreadsheet program. The compiled spreadsheet can be distributed at low cost, to staff and clientele who do not own the much more expensive spreadsheet program.

The spreadsheet is the most widely used computer application, according to Future Computing (Kaplan, 1986). Within Extension, 80 percent of computer program needs can be addressed by using spreadsheets with templates. Within the Southern Region (14 Southern states plus Puerto Rico and the Virgin Island), Extension uses six different spreadsheet programs. The use of standardized development and documentation guidelines expedites the conversion of templates from one spreadsheet to another so that more states can use a particular template. Many of the states are using the built-in conversion facilities of the spreadsheet they use or Convertacalc to translate a program developed in another state.

---

[1] Southern Regional Computer Coordinator
312 Hoke Smith Building
University of Georgia
Athens, GA 30602

Many consultants consider templates to be the wave of the future since they save large organizations and individuals from reinventing the wheel each time an application is needed. As personal computer sales to educators rise, an explosion in the use of spreadsheets and templates can be expected. Educators will be producing templates at a furious pace. Some of the templates produced may be a dream come true, making work for the educator much easier. Other templates may turn into dreaded nightmares after important and expensive decisions have been made and their calculations are found to be erroneous. The potential for error by novice template developers is immense.

Three factors make spreadsheet templates excellent tools for proactive problem solving. The factors are: 1) speed and accuracy with which the spreadsheet can work; 2) flexibility allowed the program user; and 3) the ease of template development. Proactive problem solving, careful analysis of the situation with identification of possible problems, may lead to problem avoidance and identification of opportunities for future action. Using the spreadsheets and templates as an analytical tool, past data for trends can be analyzed to predict the future with "what if" scenarios. Much creativity is involved in trying to see as many alternatives as possible.

The sections of the paper that follow discuss: 1) the advantages and disadvantages of using templates; 2) problems which arise in using templates; 3) guidelines for designing, programming and documenting easy-to-use templates; 4) procedures for auditing and reviewing templates; and 5) suggestions for training others to use templates with clientele.

Advantages of Using Templates

Templates have not been a great success to software publishers, but they are successful within large corporations and institutions because their applications involve many users and demand standardization. Unlike commercial, generic templates that don't fit anyones needs, in-house templates are designed for specific problems.

Within Extension, spreadsheets have been used as a simulation or problem solver for business and family matters, as an educational delivery tool, and as an information management tool. Most spreadsheet users began by setting up a table for a quick-and-dirty presentation of data, then moved on to more complex tasks using "what if" statements and formulas.

Templates developed within an organization can be used to standardize policies and methods, such as making sure members of the organization apply Generally Accepted Accounting Principles or convey expertise. A template can capture the expertise of a particular staff member, such as a subject matter specialist, of an organization. The expertise can then be passed around, thus raising capabilities of members of the organization. Templates close the gap between the subject matter knowledge of an expert and the knowledge of computers and programming.

A template is more than rows and columns, it is a model of a solution. Most templates model reality as carefully as possible. In modeling and simulations, everything that can be done with a calculator can be done better with a spreadsheet. The automated features take the drudgery out of modeling and allow the user to try different approaches, playing through scenarios. Many powerful models can be built using the simplest of math and logic functions. Spreadsheets are an all-purpose tool. Templates don't have to be complicated to be useful. The user, by building his own model, can better understand the reasoning behind what is happening. With spreadsheets, nothing is cast in concrete. The first uncomplicated model can evolve into more sophisticated models.

According to Software Access (Foster, 1985), many people use spreadsheets when off-the-shelf applications are available to them. One reason people choose spreadsheets over specific application programs is they have more faith in the numbers if they know how they were produced, and can modify the program to produce the output they require. A library of templates can probably address any needs an individual user or organization might have, with only a minimum of learning time. According to Schrage (1984), templates are a bridge to make software more immediately useful to a computer user.

Disadvantages of Templates

Disadvantages of templates are seldom emphasized but several have been noted. 1) Spreadsheets and templates, advertised as easy to use, in reality require skill and knowledge of logic for high quality programming. 2) Templates require a significant amount of time to develop. 3) In many cases the printout of a computer spreadsheet is taken as "gospel," with the formulas and the data entered into the template going unchecked. 4) If erroneous templates are passed on to others, mistakes can spread like an epidemic. 5) The non-technical user of spreadsheets does not seem to be aware of the critical need to check data and to establish guidelines for the use of templates.

The automation of spreadsheets has contributed to some misconceptions. Beginning users may become intimidated by powerful functions borrowed from accounting, and statistics and complex commands needed to use spreadsheets. If the user/programmer has an understanding of the underlying formulas such as IRR (internal rate of return) and NPV (net present value) and their limitations, then they will not be fooled by some analyses. Spreadsheets can be an aid to financial understanding, but not a substitute for it.

The most frequent template errors occur in formulas - specifically ranges. Several other common problems (Freeman 1984) include: 1) logically inconsistent models; 2) conceptually flawed models; 3) inconsistent data input format; 4) use of the wrong template for the task; 5) use of templates developed by others without knowledge of the assumptions and purposes for which it was developed; 6) unknown changes made in templates passed on to others; and 7) undocumented templates (Freeman 1984).

Spreadsheets with templates are fine for "personal decision support employed by individuals or small business," but may be inappropriate for decisions made by large conglomerates with many operating subdivisions, according to Burkan and Wark (Pellate 1985). Microcomputers using spreadsheets still lack all the power necessary to effectively handle the accurate consolidation of financial statements from organizations with many divisions. This disadvantage may change over the next few years as more powerful microcomputers linked to mainframes reach the desks of decision makers within large organizations.

The more sophisticated spreadsheets may be a trap to subject matter specialists as they try to program templates. The most frequent trap occurs when using more and more complex macros to incorporate error trapping in the template. In many cases, it probably would be better to use another computer language rather than the spreadsheet. Complex macro-driven templates are harder to modify than programs written in other languages like C.

Finding Good Templates

Walden (1984) suggests looking for the following attributes in choosing a template: 1) Applicability - choose a template that performs as close as possible to the task that needs to be done, modification takes time and money; 2) Suitability - choose a template that performs calculations in a manner consistent with and acceptable to your organization; 3) Expertise - purchased templates give the opportunity to capture the expertise of a truly knowledgeable practitioner; 4) Sophistication - sophisticated templates are designed to be easier to: use, learn, and understand because of better documentation. The layout of a sophisticated template should help the user understand the logic of the calculation. Better control of data entry and error-trapping procedures are designed into more sophisticated templates. For a template or spreadsheet program to be an asset, it should receive excellent marks in the following categories:

User Interaction
*Easy to use
*Commands lead to predictable results
*Easy-to-read displays

Documentation, Training and Tutorial
*Documentation adequately describes operation
*Index/Table of Contents of sufficient size to help inexperienced user
*Tutorials or Example Entries
*Training aids available
*Technical support available

Features
*Built-in calculations
*Appropriate for type of work
*Performs as claimed

When to Avoid Spreadsheets and Templates

Spreadsheets are not the universal answer to all numerical problem solving. In very simple cases it is easier to use the back of an envelope, a pencil and/or a hand-held calculator. In very complex cases, some spreadsheets may not be designed to be used effectively when the problem contains a large number of records that must be located and revised frequently (i.e. invoices, bills, checks, and other types of transactions).

DEVELOPING AND DOCUMENTING TEMPLATES

The Southern Region has adopted a set of guidelines for developing and documenting templates. The guidelines adopted by the Southern Region were adapted from the guidelines developed by the Pennsylvania and Minnesota Extension Services.

Every organization should develop or adapt template development and documentation standards/guidelines. Guidelines should be designed in a manner so that they are easy to use and understand. Guidelines can be developed within an organization so that all templates are standardized - they are similar in appearance, input and output are identified and formatted in the same manner on screen and on a printout.

Developing a Template

A summary of the guidelines for developing a template follows:

1) Become familiar with the computer and its operating system through hands-on experience.
2) Completely answer the development questionnaire. Quantify or define the problem or idea. Ask if helpful information can be generated by developing a template. Does the staff have the expertise to develop the application.
3) Use a team approach to development. Bring together the best experts available - a subject matter specialist, a spreadsheet software consultant, and a developer (if the subject matter specialist is not the developer).
4) Secure the organizations design and documentation, if available. Use these and other guidelines to assist in the development of a clear and concise program with complete documentation.
5) Design the template for ease of use and clarity, using team members for advice when appropriate.
6) Make assumptions a highly visible part of the template so they can be questioned and, when needed, changed.
7) Templates should be tested, reviewed and edited by the team after development and before the documentation process begins.
8) Document the template using the organizations guidelines or those within this paper.
9) Test the template and documentation with the development team, making editorial changes as necessary. Test the results against an example that is correct.
10) Field test the template and documentation with three reviewers. Have the reviewers, department head and head of computer services sign-off on the template before distribution.
11) Review suggestions made during the field test, then make final revisions before the template is distributed. If distributed outside the institution, the template should contain copyright and disclaimer notices.

Tips For Developing Templates Include:

Start with the results of the development questionnaire and a paper design. A good design can save hours of work and potentially gross errors.

1) Identify the Template - Choose a meaningful and helpful name. Identification of the template includes the template title, author, programmer, organization name, version number, and copyright on the first few lines of the template. This is the first section the user sees.
2) Format for the Screen and Print - Identify input, calculation, and result areas of the template. Make data input and results areas resemble existing forms. Long templates may need a screen table of contents. The manner in which the template is formatted on the screen will depend upon the application. Because of the way the spreadsheet looks on the printout, a choice between a visually pleasing screen or printout may be necessary. Choose the medium that is most important to the user. Utilize the space on the screen, making inputs run vertically down the page as one normally reads, rather than skipping all over the spreadsheet.
3) Use uniform input cells - Visually lead the user to the input area. Use an entry path or arrows pointing to the entry cell.
   [Ex. ----->> <<-----, >>>> <<<<]
4) Isolate Variables - Group key parameters in one area of the template. Have all references to that variable address a single cell. This procedure will simplify changes, reduce errors and make spreadsheets more readable. Enter data in either rows or columns, not both.
5) Focus on Key Ratios - Plan the spreadsheet around parameters that drive the problem. These are the manipulated values or the numbers on which the manipulated values have impact.
6) Use English - Take a few extra keystrokes to make the spreadsheet easy to read. Use English, not jargon or abbreviations.
7) Keep It Simple - Long equations are difficult to understand. Use intermediate variables to break up equations. Errors in long equations are hard to trace and their logic is difficult to decipher. Do not mingle data entry with calculations. Stream-of-consciousness written macros and equations are hard to modify or document. Structured programming can help unscramble and organize macros and equations.
8) Use Modules - An entire spreadsheet doesn't have to be built at once. Create a working spreadsheet for one month, or one topic, then copy that to other sections of the spreadsheet. Finding an error in logic is easier in a small subsection of a template.
9) Use Hash Totals or Crossfoot - Add numbers both horizontally and vertically across columns to double check accuracy, if applicable.
10) Use visible assumptions - Make assumptions stand out from hard numbers.
11) Check Equations - Check the logic of equations and the typing. Then recheck. Use an auditor program such as DocuCalc to examine the formulas of the template.
12) Save Models - Save copies of the model as it develops and as changes are made. Always have backup copies.
13) Protect Integrity - Use the cell protection feature or compile the template to protect the integrity.
14) Build in Checks - Use If statements, macros, comparisons with known ratios - anything that can be built-in to check and safeguard against errors in either data entry or logic.
15) Recalculation - Use manual recalculation when entering data into large spreadsheets to speed entry. Be sure to inform the user that automatic recalculation has been turned off.
16) Printing and Graphing - Instructions should be clearly noted on the template and in the documentation on how the user can print or graph a hard copy of the results.
17) Have Fun - One of the major payoffs of building a template is the process itself - the understanding gained of the problem and the creativity used to solve the problem.

Documentation Guidelines

Documenting a spreadsheet template may take many hours, but each hour pays off handsomely as other users can pick up the documentation, learn the assumptions of the template, and begin using it within a short time frame.

The Southern Region has adapted guidelines and worksheets for developing documentation. The worksheets are available in printed form or on disk. The disk version can be completed with a word processor. The worksheets include the following documentation sections:

Title page
Table of contents
Program overview and acknowledgments
Objectives and capabilities
Audience
System requirements and support contacts
Interpretation of results
General instructions for using the spreadsheet
Other supporting documents
Management procedures (backup copies, etc.)
Template design
Template input descriptions
Data input collection sheet
Assumptions/limitations of calculated values
User tutorial
Return form for noting errors and suggestions

REVIEW PROCESS

A set of review procedures, in combination with template auditing software, can reduce the possibility of errors within a template. As spreadsheets grow in popularity as educational tools, they are passed from one person to another. Modifications are made, to fit the users problem, with each set of hands a template reaches. In many cases as templates are passed on to others, they are used improperly. One way to protect users and developers from improper changes is to select a spreadsheet that has a lockout feature with a password, or to compile the template with a program such as Baler.

The only users who can modify the template are those having the password or the source code.

Freeman (1984) suggests a set of straightforward steps for auditing spreadsheets. The steps include: 1) Insist on an audit of all spreadsheets used for important decision making; 2) Create and enforce an audit trail; 3) Establish responsibility for the model; and 4) Require an audit each time the model is modified. Expanding upon Freeman's guidelines, a seven-step review and audit procedure is advocated.

1) Set Standards - Develop a strategy for the development of personal computer applications within your corporation or institution. Use an applications checklist to assess whether the application is appropriate to the personal computer and to a spreadsheet. Set standards for development that encourage consistency and reduce confusion.

2) Review and Audit - Insist on a review and audit of all templates that are used for important decisions or that are passed on to others.

   a. Use hashing or crossfoot techniques, totaling across rows and down columns to check calculations. Spreadsheets created on paper always add numbers both horizontally and vertically to double-check for accuracy; automatic spreadsheets should be no different.

   b. Verify the formulas within the template. Use commercial software programs that can audit templates. Even a small error in a template can lead to wrong conclusions that could cost time and money. Audit software programs can print out the formulas in the exact format of the spreadsheet, wrapping around long formulas into a single column. The programs provide a backup paper copy of the template. The two-dimensional printouts of spreadsheet formulas make the verification of the internal structure of templates routine.

   c. Give the template to three persons for blind review. Prepare a sign-off sheet for signatures that must be obtained before distribution.

3) Create and enforce a review process audit trail. If the template has been developed according to guidelines noted above, much of the audit trail has been built-in. In addition to the printout of formulas listed in Step 2, the author's name, the programmer's name, the date created, the date of the last update, the type of input and output that are expected, and the template's documentation designed according to the guidelines are part of the audit trail.

4) Establish responsibility for the template. A master copy of the template, all documentation and the printout of all formulas should be kept in an archive file held by the developer and the computer coordinator. Changes in the template should be made by authorized personnel only. A list of all persons having copies of the template should be kept in case revisions of the template are necessary.

5) Audit and Review Revisions - If, after use, a refined or enhanced version of the template is desired, the refined template should go through the same development, programming, documentation and auditing procedures as the original.

6) Establish a distribution procedure for within the organization and for distribution to other organizations. As noted above, a list of all persons having copies of the template should be kept. Procedures for distributing templates should include statements about not copying templates for others. If copies are desired, they may be obtained from the archive copy to insure that no modifications have been made. The author of the template, or the organization, may wish to copyright the template and documentation as one deterrent to unauthorized duplication. Compiling the template is another deterrent to modification.

7) Set up a procedure for collecting suggestions for changes to a template or for development of entirely new templates.

TRAINING OTHERS

As templates are distributed within an organization, it is essential that all potential users are adequately trained. McLaughlin (1984) has developed a list of considerations designed to aid in the conducting of a training session. A summary of considerations follows:

1) Establish specific objectives for the session.
2) Tailor the session to the audience.
3) Limit lecture format, use participatory learning techniques.
4) Plan room setup based on group size and equipment.
5) Keep ratio of participants to computers low.
6) Consider staffing requirements, location, power supply, layout, and setup.
7) Have handouts, reading lists, and other information displayed.
8) Promote users groups and networking among participants.

At the end of any training session, take a moment to pass out evaluation forms. These forms should include questions on the users' evaluation of the template(s) and on the training session itself. Compile the information gathered and make changes as necessary in the template(s) and/or training procedure.

APPLICATIONS

Applications templates designed and documented according to the Extension guidelines are just becoming available for several spreadsheets and models of computers.

SUMMARY

In summary, several actions can be implemented within an organization to make sure spreadsheet templates are carefully developed and used so they do not become a nightmare. By carefully following planning, programming, and documenting guidelines, and training staff, organizations can have templates that: 1) are not misused and 2) remain relatively free of error. It may take more time in the beginning to establish guidelines for templates, but time and money will be saved in the final analysis.

TRADEMARKS

The use of trademarks or tradenames within this paper does not constitute an endorsement by the Cooperative Extension Service. The following are registered trademarks of their respective companies:

ConvertaCalc - Microdecision Systems, P.O. Box 1392, Pittsburgh, PA 15230.

DocuCalc - Microdecision Systems, P.O. Box 1392, Pittsburgh, PA 15230.

Baler - Brubaker & Associates, Inc., 116 West Main, Delphi, IN 46923.

Lotus 1-2-3 - Lotus Development Corp., 55 Cambridge Parkway, Cambridge, MA 02142.

Multiplan - Microsoft Corp., 10700 Northrup Way, P.O. Box 97200, Bellevue, WA 98009.

Supercalc - Computer Associates International, Inc. Micro Products Division, 2195 Fortune Drive, San Jose, CA 95131-9990.

Contact: Margaret P. Ezell
Southern Regional Computer Coordinator
312 Hoke Smith Building - UGA
Athens, GA 30602

404-542-5119

REFERENCES

Berry, T. Writing Structured Macros. Lotus, August 1985, pp. 64-67.

Extension Computer Services, The Development Division. Multiplan Guidelines for the Macintosh, Version 2.0. University Park Cooperative Extension: The Pennsylvania State University, 1985.

Fawcette, J. E. Using Spreadsheets to Model Your Business. Personal Computing, December 1984, pp. 70-82.

Foster, E. Building Simple Spreadsheets. Personal Computing, January 1985, pp. 61-67.

Freeman, R.M. A Slip of the Chip on Computer Spreadsheets Can Cost Millions. Wall Street Journal, August 20, 1984, p. 10.

Grushcow, J. Avoid These Common Spreadsheet Errors. Lotus, July 1985, pp. 59-62.

How Personal Computers Can Trip Up Executives, Business Week, September 24, 1984, pp. 94-102.

Kaplan, J. Some Future Trends in Spreadsheets. Lotus, February 1986, p. 138.

Karten, N. Developing Applications for Others. Lotus, December 1985, pp. 77-80.

Laric, M. V. and Stiff, R. VisiCalc for Marketing and Sales. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1984.

McLaughlin, E.C. Considerations for Conducting a Microcomputer Workshop. Journal of Nutrition Education, 1984, p. 76H.

Osgood, W.R. and Molloy, J. F., Jr. Business Decision Making for Higher Profits: A 1-2-3 Business User's Guide. Somerville, MA: Curtin & London, Inc., 1984.

Pallatto, J. Decision-Support Systems. PC Week, October 16, 1984, pp. 51-56.

Rubin, C.A. Moving Up With Spreadsheet Math. Personal Computing, January, 1985, pp. 48-57.

Schrage, M. Templates Are the Bridge for Troubled Program Users, Washington Post, May 7, 1984, p. 13.

Sharpe, W.F. Electronic Spreadsheets. Financial Analysts Journal, July/August, 1984, pp. 15-20.

Walden, J. The Template Question. Business Computer Systems, December 1984, pp. 62-64.

ADCIS Proceedings, November 1986

DEFINING INTELLECTUAL PROPERTY RIGHTS IN THE DISTRIBUTION OF SOFTWARE

Margret Hazen, University of North Carolina at Chapel Hill

Thomas Lee Hazen, University of North Carolina at Chapel Hill

ABSTRACT

Implications of the alternatives and constraints of the various types of legal protection of software -- copyright, patent, trademark, trade secret, and contractual -- are analyzed in the instructional software development setting. No single component of intellectual property law completely protects both the software developer's ideas and their expression. Accordingly, the nature of a software development project and its goals determine the best methods of protection.

I. Introduction: Legal Issues

Intellectual property rights can be protected within two frameworks--one as defined by statutes and the common law, and the other by arranging private individualized contracts (Davidson, 1983). Copyright, patent, and trademark protection are three basic theories used to protect intellectual property within statute and common law. Alternatively, the law also recognizes two ways in which an individual can protect his or her own ideas: by taking appropriate steps in creating and securing trade secrets and/or by entering into contractual agreements (e.g. licensing arrangements) to protect ideas and processes. This paper analyzes the implications, the alternatives, and constraints of the various types of legal protection of software intellectual property rights within the instructional software development setting.

II. Statutory Methods of Protecting Intellectual Property

A. Copyright protection

The copyright laws protect artistic works and thus protect the written embodiment of the program (i.e. source code and object code) as well as the designs, words and configurations as they appear on the screen (Brooks, 1985). The federal copyright statute not only forbids the unauthorized copying of copyrighted works, but it also prohibits the unauthorized creation of a work that is a derivative work of the copyrighted work.

However, the copyright laws are limited to "means of expression" or to the delivery of the idea and thus do not extend to the protection of ideas per se (Conley & Bryan, 1985). For example, the idea of a computer maze game is not protectable under the copyright laws. The expression of that idea, that is, ghost monsters that gobble power dots, is however eligible for copyright protection.

Although unauthorized derivative works are illegal, the term "derivative work" has not been firmly defined. Using analogies in the fine arts, two examples of works defined as derivative are: (1) the musical version of a dramatic play, and (2) the movie based on a book. In the context of software, it is very difficult to decide at what point the borrowing of an idea from another work is so similar to the original that it becomes a derivative work. For example, is a spreadsheet program -- written in different code but having the same basic screen format and requiring the same keystroke commands as an earlier copyrighted program -- a derivative work of the copyrighted program? Both limiting the protection of an idea to the means of expression, and the lack of a definition of a derivative product weaken the copyright law's ability to protect the software developer.

B. Patent protection

The patent laws protect ideas but only to the extent that they are useful and are novel in application. The United States Supreme Court has held that programming code (i.e. algorithms) and programming techniques alone cannot enjoy the protection of the patent laws. Accordingly, the patent laws do not play a significant role in the protection of instructional software (Conley & Bryan, 1985).

C. Trademark protection

The trademark laws protect trade names and logos that identify a particular product with its manufacturer, producer, or supplier. The trademark laws can be used to protect program names, logos, and other identifying features, but will not protect the program code or the ideas that contribute to the program. For example, if a finished program has unique screen designs and characters for its title screen or name of the lesson, it is possible to establish trade mark protection for these features; but trademark law cannot cover each screen of the instructional software program.

III. Self-Help Methods of Protecting Intellectual Property

A. Trade Secret protection

Ideas and processes (such as programming techniques) can be protected so long as the owner keeps them a secret (Gilbourne & Johnson, 1982). This protection will be recognized even if the ideas and processes are neither copyrightable nor patentable. Among the advantages to trade secret protection are the unlimited duration, and the absolute protection accorded to properly kept secrets. The major disadvantage is that trade secret protection depends on secrecy which, in turn, means securing the program with stringent copy protection schemes and the use of encrypted code.

Another disadvantage is the fact that reverse engineering is permissible for processes that are not patented, even if the expression of the ideas is protected by a copyright. For example, the Coca Cola formula is a trade secret and is not protected by patents. If you can chemically analyze Coca Cola and derive the formula, you can use it in competition with Coca Cola. However, you cannot legally use the Coca Cola trademark or any copyrighted advertising material in distributing the soda based on the Coca Cola formula.

These issues have particular implications for software distribution. So long as programmers or designers using reverse engineering techniques on a program do not use the same programming code but instead derive their own code (although based on someone else's ideas), they can change the words and use subroutines and menus without infringing on that person's rights. The increasing sophistication of disassemblers and deciphering techniques makes reverse engineering a potential threat to programs protected by compiled or encrypted code. Since algorithms are not patentable and reverse engineering is thereby permissible, trade secret law is limited by the increasing sophistication of software engineering.

Perhaps the biggest problem with relying on trade secrets in the academic environment is that the security protection and secrecy necessary to give both legal and actual protection to trade secrets runs counter to the academic ideal of the free exchange of knowledge. The same may be true for restrictive licensing arrangements (Crecine, 1985).

B. Contract protections

The law recognizes contractual protection of intellectual property. Such contractual provisions frequently take the form of licensing agreements. Licensing agreements limiting the use of the program (for example, to one user or one machine for each package paid for) can be useful and are frequently necessary as a supplement to copyright protection (Vale, 1985). Limited use and dissemination agreements may also be necessary to protect trade secrets when sharing the secret process, formula, or algorithm with third parties (Milgram, 1985). Agreements supplement intellectual property protection when using copyright or trade secret law.

Relying on contractual limitations or licensing agreements combined with either copyright or trade secret protection creates problems relating to record-keeping, policing, and enforcement of the agreements unless the software is securely copy protected. Easily copied software is impossible to keep records on or police.

In defining what constitutes infringement, the copyright laws prohibit not only the copying of a copyrighted work, but also the creation of an unauthorized derivative work. Even apart from the copyright laws, contractual limitations on use may refer to the problematic concept of what is a derivative work. Consider the example of Professor Brilliant who writes an instructional program while in the employ of Learning University -- the owner of the copyright under a royalty-sharing arrangement. When the Professor decides to write part two of the copyrighted work she may be creating an unauthorized derivative work unless the contract she had with Learning University gives her the right to make future instructional modules without infringing on the university's copyright (Peterson, 1985). Similar problems arise when the author of a software program enters into a publishing contract and assigns the contract to the publisher. The author may have assigned his right to create derivative works unless there is a specific provision to the contrary in the contract.

As noted above, restrictive licensing arrangements may be in conflict with the academic ideal of knowledge dissemination. It may be possible, however, to share knowledge and at the same time to protect the author's proprietary interests. For example, a licensing agreement might provide that the licensee (i.e., each user) agrees not to engage in reverse engineering or otherwise appropriate the ideas and techniques embodied in the licensed piece of software. However, such restrictions are only enforceable if software is licensed as opposed to sold (Samuelson, 1985) and it is very difficult to enforce licenses in the context of mass distribution.

IV. Delimiting Variables for Determining Optimal Method of Protection

Because of the law's shortcomings in protecting intellectual property relating to computer software, maximum protection can only be obtained by the copyright laws combined with copy protection and code encryption safeguards. While trade secret protection or licensing arrangements can supplement the copyright laws, enforcement of these rights is problematic. If the author's desire to make the program modifiable precludes encryption, then the author's protection will be sacrificed. Similarly, deciding not to use a copy protection scheme, will increase unauthorized copying. Thus, the author's goals for, and intended use of, the software will necessarily affect the author's ability to protect their proprietary interest.

A related problem concerns the rights of employees who work on software development. For example, to what extent are programmers' techniques and routines their property to take when they leave the employment setting? (Peterson, 1985). Contractual provisions can be used to define the respective rights.

V. Conclusion

The software developer's goals and planned uses for the program determines the means of available intellectual property protection. Different approaches to both development and the method of protection may be appropriate when software is developed for distribution locally (free), in the academic community at large (at cost), or commercially (for profit). The relatively high costs and complexity of maximum protection may not be appropriate for software developed for academic or local use.

The likelihood of creating future derivative works must also be considered in establishing plans for both the development and distribution stages of any instructional software project. What constitutes derivative works and who has the right to produce them may need contractual definition when software is developed for profit.

References

Brooks, D. (1985, Summer). Copyright and the educational uses of computer software. EDUCOM Bulletin, 20(2), 6-13.

Conley, J. & Bryan, R. (1985, Summer). A unifying theory for the litigation of computer software copyright cases. Computer Law Journal, 6(1), 55-118.

Crecine, J. (1985, Summer). The role of universities in knowledge dissemination: Site licensing is secondary. EDUCOM Bulletin, 20(2), 2-5.

Davidson, D. (1983). Protecting computer software: a comprehensive analysis. Arizona State Law Journal, 1985, 611-783.

Gilbourne, M. & Johnson, R. (1982). Trade secret protection for software generally and in the mass market. Computer/Law Journal, 3, 211-272.

Helm, V. (1984). Software Quality and Copyright: Issues in Computer Assisted Instruction. Washington D.C.: Association for Educational Communications & Technology.

Milgram, R. (1985, November). Rights and liabilities of third parties who may receive another's proprietary rights. The Computer Lawyer, 2(11), 1-5.

Peterson, I. (1985, September). Bits of ownership: Growing computer software sales are forcing universities to rethink their copyright and patent policies. Science News, 128, 188-190.

Project Athena, Massachusetts Institute of Technology. (1984). Commercialization of Educational Software: Issues on Development, Marketing, and Distribution (Report No. TR9-1-85). Massachusetts: Massachusetts Institute of Technology.

Samuelson, P. (1985, December). Creating a new kind of intellectual property: Applying the lessons of the chip law to computer programs. Minnesota Law Review, 70(2), 471-531.

Vale, S. (1985, August). Practical and legal issues relating to the marketing of microcomputer software by means of site licenses. The Computer Lawyer, 2(8), 1-9.

Contact: Margret Hazen
Instructional Software Development
Microcomputing Support Center
232 Davis Library 080A
University of North Carolina
Chapel Hill, NC 27514

919-962-0101

ADCIS Proceedings, November 1986

PROTECTING YOUR DEVELOPMENT INVESTMENT:

THE PROJECT SCHEDULE AS A MEANS TO CONTROL RISK

S. M. Leslie

International Business Machines

ABSTRACT

Whether developing a course using only computer created text and graphics, or a more complex course using AI techniques or videodisc, the cost of development is increasing. Increasing cost implies increased risk of not realizing the appropriate return on the development investment. Managers need a methodology for assessing risk. This paper presents a development model and its associated deliverables. The deliverables are measured against a set of gating criteria to assess risk. Review by various levels of management is discussed.

INTRODUCTION

As teachers and educators working in the field of Computer Based Instruction, we are faced today with a mixed blessing. The tools available to us allow greater creativity and more effective instruction than many of us who grew up "lecture oriented" dreamed possible. But the very power of those tools, and the resulting complexity of the instruction, carry a price tag. We are producing instruction with a "shelf life," something the classroom lecture never had. The cost is well justified, but it is higher. The people are more expensive, and usually the production is more expensive. As a result, today's managers must closely control their projects, for the investment they risk is not only sizable but more easily quantifiable and highly visible.

The value of using visual images in instruction is certainly not news; we really did know even before TV. But the ability to deliver visual images, to achieve all levels of symbolic transfer of information, is new. The book combines written symbols (words) and pictures. Film combines motion sequences and audio. Traditional Computer Based Instruction (CBI) and Computer Based Training (CBT) provide individualization and assessment capabilities. Today we have the technology to combine all of these into Integrated CBI/CBT. Today's courseware can use text, animated or still graphics produced in the PC, synthesized speech (from the PC), motion sequences or still pictures with or without audio from the videodisc, or combinations--like PC graphics overlayed on video.

This paper will use the terms media treatment and media technique interchangeably to mean a specific media used in a specific way. Media means the source of the material and includes computer programs, videodiscs, and auxiliary electronic equipment like oscilloscopes. Treatment includes text, graphics, animated graphics, motion sequences, and stills. Note that animated graphics, a technique, can be delivered from a program or from a videodisc.

Management of the risk associated with developing this type of courseware requires a development model.

A DEVELOPMENT MODEL FOR INTEGRATED CBI/CBT

The many excellent models used today for CBT development quite appropriately owe their heritage to software development models and instructional system design models. Most of them are very similar, differing only by how the steps are collected into phases and the names given to the phases. They are all good models, just different flavors.

These models, however, were developed to support text based, "green screen" CBI or CBT. In today's world--and more importantly--in tomorrow's world, instruction will be more complex from both an architectural and a creative point of view. Courses will use media techniques which must be integrated if the instruction is to be "a course" rather than a group of "neat" instructional sequences stuck together like a Rube Goldberg machine.

Today's PC based courseware uses color graphics including techniques previously available only in film, like animation, wipes, and fades. Leading edge companies and institutions are forging ahead into intelligent courseware. All of this is with a "plain old every day" personal computer.

Today's courseware integrates motion sequences from videodisc. Other external devices are also being attached (through the standard RS-232 port) and--under control of the PC instructional program--being used for training in a variety of domains including physical manipulation (psycho-motor) skills.

In order to efficiently handle this complexity and utilize media treatments effectively, the traditional models need to be extended. Figure 1 shows such an extended development model. It allows for the delivery vehicles in use or in R&D today, and is extendable to those we do not yet visualize. This model can be adapted for a specific course by retaining those legs which apply to the course and eliminating the others. Using this model will also facilitate development of follow-on courses, which add other media techniques.

MODULARIZING THE DEVELOPMENT

The model in Figure 1 uses a set of modularized activities. There are, of course, additional modular sets of activities within each block. However, this paper will stay at the level of the diagram shown in Figure 1.

Figure 1 makes the following assumptions:

- Goals of the course have been established
- First draft of the terminal instructional objectives is complete
- Business Case or other justification is complete
- Project has been funded

In addition to providing clear checkpoints, modularizing the development process provides a framework for allocation of resources to balance efficiency and quality.

Integrated education means that the pieces of the instruction must work together as an integrated whole. To accomplish this significant goal requires that the course be designed by one "master architect" or "master architectural team," clearly keeping in mind the overriding objective of using media effectively as part of a grander plan. Therefore, the process starts with two sequential steps.

If the course is to be truly integrated, decisions about the course as a whole must be made before decisions about media treatment used within the course. The instructional objectives, while certainly suggestive of specific media treatment, are developed first, before media treatment is decided. Often multiple media techniques will be used within a single objective, even at the enabling objective level. The most common error is to overuse the expensive media. Even topics which clearly call for motion sequences, like management training subjects, have considerable content which is better delivered in text or computer graphics.

Once the master plan (high level design) is in place, including global decisions about where in the course to utilize various media techniques, the development effort can be split among specialized teams. There still must be a master architect in place to work with the teams, to recognize when the teams need to interact and to facilitate that communication.

Splitting the effort not only allows for a shorter development cycle; it also supports quality. If these pieces are developed sequentially, the spontaneity that makes them work well together is often lost. (The excitement about one piece can be lost by the time later pieces are being developed.) The emphasis on a "control program" allows a programmer to focus on making the pieces all fit together correctly, while leaving the other teams free to concentrate on smaller content related sections of the course.

The work products of these teams are then integrated, by "plugging them in" to the control program, to produce a course which is:

- Instructionally effective
- High in quality
- Cohesive and consistent

DELIVERABLES AS DECISION TOOLS

Managing an Integrated CBT/CBI project means delivering a quality course, on time and within budget. The manager's task is always to minimize the risk that one of these targets will not be met. Projects will differ in the relative importance of the factors; but each will always need to be monitored. The project deliverables provide the best means of minimizing the risk of an unsuccessful project.

Deliverables can be measured against the following criteria:

- Supports the instructional objectives
- Is consistent with the estimated size of the course
- Shows a clear understanding of the content material
- Shows creativity in use of media
- Is consistent with budget
- Was produced in the time allocated

If the deliverable fails even one of these gating factors, the project should be reevaluated. Frequently development is continued because "we've already spent so much money (or time) on it." The project was originally funded because the investment was cost-justified. If the deliverables indicate that the finished course will seriously fail one of the above gating factors, then either the benefits expected will not be realized or the cost will be greater than anticipated. In either case, the appropriate business judgment can only be made by reevaluating the project's business case in light of the current status.

Usually the project can be redirected or revitalized with staffing adjustments. Catching a problem early makes this redirection possible. Frequent checkpoints catch problems early.

However, every reevaluation should include consideration of discontinuing the development. Occasionally the risk of producing an unacceptable or ineffective course is so great that the best business decision is to stop the project.

DELIVERABLES

Each activity in the project plan has two steps: a planning activity and a development activity. Figure 2 shows the deliverables from each activity. Names are shown for the deliverables, but don't take the definitions of these names for granted. Each organization, each vendor, has their own format and name for the various deliverables. And there is by no means a consensus on the use of various terms. For example, a "script" can mean:

- A traditional video and audio script
- An English-like list of steps to be implemented in computer code
- A document including both of the above
- An audio/video script with narrative describing the CBT activities

There is someone out there who will read that list and say to themselves, "But the most obvious definition isn't on that list!" And if you think that one's bad, try "storyboard."

It's not the names that are important; define the contents. And do it before beginning in-house development or contract work with a vendor. A definition could be, for example, a bulleted list, a narrative description or an outline showing section headings.

REVIEWS

Reviews by different levels of management each take a different form, even though they are reviewing the same deliverables. The team leader, who may or may not be a manager with personal responsibility, is in daily contact with the work. This course should be the team leader's only responsibility. If the course is a small one, the team leader may be one of the developers. Even with that level of daily review, the team leader should, with some level of formality, hold a review at each checkpoint and make a go/no-go decision or recommendation.

The team leader's manager will most likely have responsibility for multiple training or instruction projects. CBI or CBT may be a new and unfamiliar area of education, or the manager may manage multiple CBT/CBI projects. The manager needs to know not only the status of the project (deliverables on time and consistent with the gating criteria), but also needs to review the major deliverables. Whether this is a substantive or cursory review will depend upon the manager's style and technical background.

The next level of management most likely has responsibilities that go beyond training or instruction. This "manager of managers" may, for example, have overall personnel responsibility or may be a Dean or a Curriculum Director. This manager needs only the reassurance that the project is on target and needs to be alerted to possible problems. Figure 1, labeled to show completed activities and schedule dates, makes an appropriate review document. Any problems would, of course, require additional documentation.

PASSING THE BATON

Each deliverable feeds the next activity in the process. An ideal world would have such crisp and definitive communication that a paper document could be handed from one work group to the next with no further communication necessary. Unfortunately, this is far from an ideal world.

In a small project, the entire team could participate in each activity without creating a "mob scene." But it doesn't take a very large team before there is a critical need to limit the number of people participating in the creation of work products. The problem of a smooth movement of the course development between teams should receive significant management attention during the development of the schedule. Adding reviews--and adding people to reviews--adds time.

SUMMARY

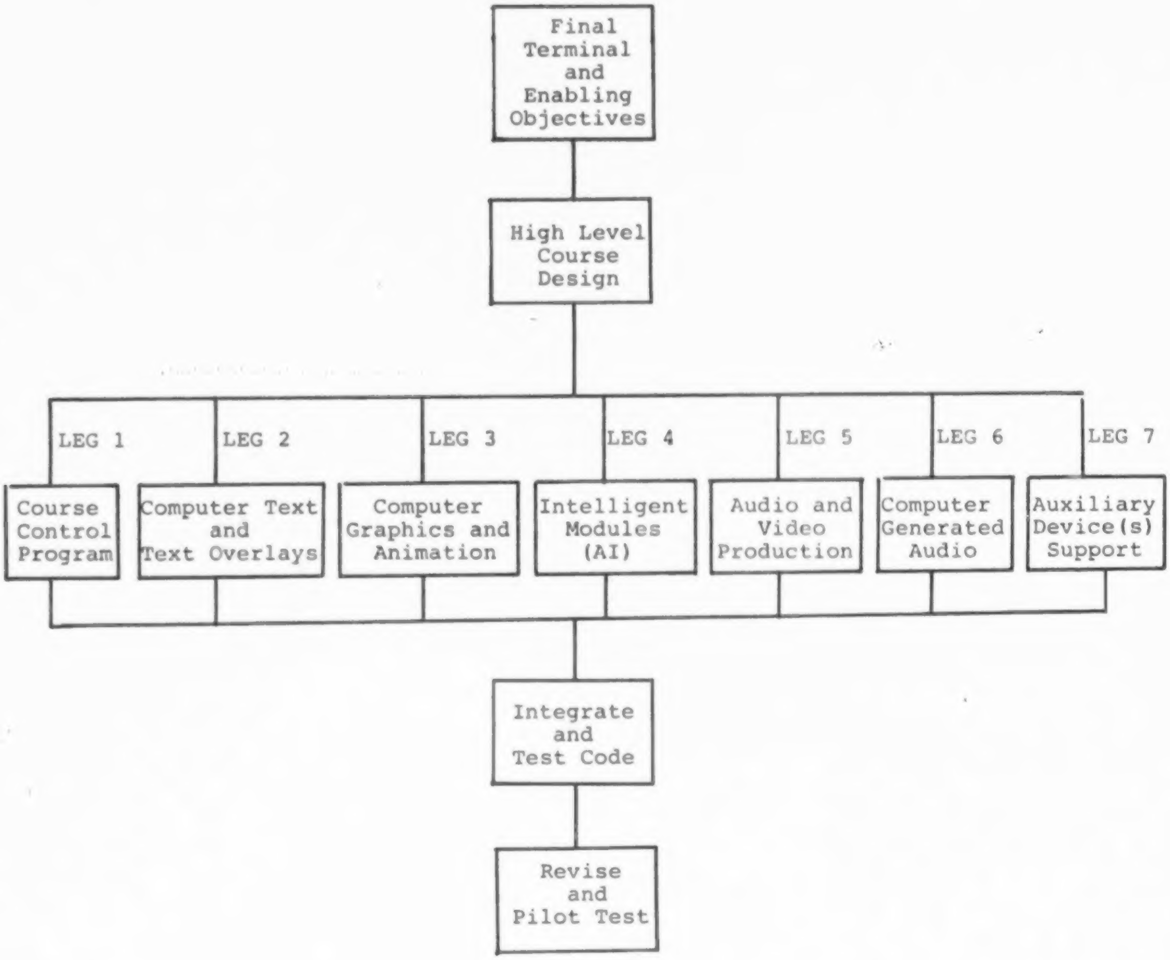
The model discussed in this paper provides a vehicle for appropriately managing Integrated CBT/CBI projects. There undoubtedly are or will be other such models. The exact format is not important. The key is early agreement on what is to be documented and a clear set of criteria which will enable management to protect their sizable investment of dollars, people, and creative commitment.

Contact: S. M. Leslie
IBM Corporation, 14-07-00
5601 Executive Drive
Irving, TX 75038

214-556-4791

FIGURE 1

A MODEL FOR INTEGRATED CBT/CBI DEVELOPMENT


Final Terminal and Enabling Objectives
High Level Course Design
LEG 1
LEG 2
LEG 3
LEG 4
LEG 5
LEG 6
LEG 7
Course Control Program
Computer Text and Text Overlays
Computer Graphics and Animation
Intelligent Modules (AI)
Audio and Video Production
Computer Generated Audio
Auxiliary Device(s) Support
Integrate and Test Code
Revise and Pilot Test

NOTE: The multiple activities occurring in the various legs are regularly reviewed by the Course "Architect."

FIGURE 2

INTEGRATED CBT/CBI DEVELOPMENT MODEL: DELIVERABLES

ACTIVITY	DELIVERABLE	COMMENTS
Objectives	1. Final Terminal Objectives	Measurable
	2. Enabling Objectives	Measurable
Course Design	3. High Level Design Document	Can be narrative, includes standards documents
	4. Final Course Design Document	Includes detailed media treatment/ techniques
Leg 1: Control Program	5. Programming Flowcharts	Show branching; multi-level
	6. Running Code	Includes unit testing
Leg 2: Computer Text	7. Script	Text is bulleted, not-narrative
	8. Running Code	Includes unit test
Leg 3: Computer Graphics	9. Graphics Design	Includes standards documents and samples
	10. Screens	Includes unit test
Leg 4: Intelligent Modules	11. Semantic Nets	Includes standards for input
	12. Running Code	Includes unit test
Leg 5: Audio and Video	13. Script	Can include linkage statements to other media
	14. Completed film or tape	Includes editing cycle(s)
Leg 6: Computer Gen. Audio	15. Script	Includes code sequences and spoken words
	16. Running Code	Includes unit test
Leg 7: Auxiliary Device	17. Design	Form dependent on device; includes standards
	18. Running Code	Includes unit test
Integrate Code and Test	19. Test Plan	Includes system test and test with students
	20. Validation Report	Identifies problems and recommend "fixes"
Revise and Pilot Test	21. Test Plan	Includes criteria for success
	22. Pilot Test Report	Recommends implementation of Release 1 (or not)

MANAGING A COURSEWARE DEVELOPMENT MANAGER: A PRIMER

Valerie E. Mock

Georgia State University

ABSTRACT

This presentation gives managers who have been assigned the responsibilty of manging a manager of courseware development a guide on how to staff, organize, plan for, and control such a person and the people who do the work. Specifically, it reviews the elements of managing any software development group and elaborates on what makes courseware development different.

INTRODUCTION

With the advent of computer-assisted instruction becoming more commonplace in industry and on campuses, many managers are now responsible for an additional task: managing a manager or group leader who manages a department or group of courseware developers. This responsibility is not significantly different from managing any development manager, but there are differences.

A REVIEW OF MANAGEMENT TASKS

Management is often defined as performing five basic functions: planning, organizing, staffing, controlling, and motivating. In developing courseware, decisions must be made regarding how to: develop the materials, organize the work, staff the project, control the project to make sure that the plans are achieved according to the schedule and within the allocated resources, and motivate the people involved to complete the task.

COURSEWARE MANAGEMENT TASKS

Planning

Planning for courseware involves deciding:

- What is going to be produced
- Who the users of the product are (the students, the instructor, and any other persons)
- What the general content and instructional goals are
- What the likely instructional environment is
- Who is going to help create the product
- What schedule is going to be followed
- What process will be used to ensure that the materials get developed effectively, efficiently, and on time

Organizing and Staffing

The skills needed to achieve instructional goals of the courseware require your manager to carefully match the required skills against the available personnel. The manager will then need to decide how to organize the team to accomplish the goal and make sure that the lines of authority and each person's role are very clear. Generally, these roles are:

- A content expert
- An instructional designer
- Developers
- A graphics designer
- A reviewer/evaluator
- An editor
- A usability expert
- Support personnel

You may want to help the manager determine if these people can be hired on a temporary basis or "borrowed" from other departments or sources as necessary.

Controlling

Control is important because you want to make sure that the courseware materials work as intended and does not produce a negative learning experience. Control of the courseware is normally established by testing the materials within the project and then with sample students before being released for others to use. Project control is often established by making sure the manager or group leader defines a development process and a schedule of events within that process, and then meets those defined events.

The phases suggested for use as a formal development process include:

Phase I. The Business Planning Phase--The purpose of this phase is to determine that the effort involved in producing the materials are worth the gain(s) to be achieved, how the courseware will be developed, who will form the group, what schedule will be followed, and what resources will be required. The goals of this phase are to:

- Generally determine the intended instructional goals
- Define the scope of effort required
- Identify the users of the courseware
- Identify the skills and personnel required
- Prepare a business case
- Select, organize, and assign responsibilities
- Identify and obtain the non-personnel resources
- Prepare a working budget
- Determine a schedule identifying start and completion dates
- Determine the educational/training needs of the team members
- Make a list of assumptions
- Begin to keep a record of all of the decisions made

Phase II. The Definition Phase--The purpose of this phase is to gather specific, pertinent information about the student population and the instructional environment, and then define the computer-based materials to meet the instructional needs of the students within that environment. Assessing the students and the instructional environment will help determine how students are to be taught, which attitudes, skills and behavioral characteristics they need to acquire, the most effective presentation style, and the amount of instruction possible within the intended instructional environment. The goals of this phase are to:

- Decide on a working, descriptive title of your course
- Identify the characteristics of the target population
- Identify the characteristics of the people who are going to work with the target population
- Describe the instructional environment
- Describe the overall content of the course and the content of all documentation
- More precisely define the instructional goals
- Define the instructional strategies
- Describe the general presntation of the content and the way in which the student will interact with the materials
- Describe why the computer was chosen as the instructional medium and if it is to be used in conjunction with any other media
- Determine approximate times for completion of the courseware by the student
- Determine the necessary equipment, software, and materials to be used in presenting this courseware
- Identify any texts to be used in conjunction with, or that may supplement the use of this material
- If this material has been adapted from material already published, list the title, publisher, and copyright date
- Re-evaluate the schedule and business case to determine if any adjustments are required and make the necessary changes

Phase IIR. The Definition Review--The purpose of the Definition Review is to examine the activities performed during the Definition Phase to ensure that: 1) they were performed, 2) they are accurate, and 3) they are complete.

Phase III. The Design Phase--The purpose of this phase is to design the organization, content, and purpose of the courseware in such a way that the people who will write the programs and documentation can translate your ideas into the actual materials to be used. The goals of this pahse are to:

- Determine the structure of the content
- Place the concepts into a logical sequence
- Plan and organize the components of the course
- Identify performance objectives

- Estimate the amount of presentation time for each lesson
- Identify the actual events of instruction
- Establish the textual and graphics information that will serve as the "script" of your courseware
- Determine the types and frequency of student interaction
- Identify motivational factors
- Prepare appropriate answer analysis procedures
- Determine the content of the intended documentation
- Determine the programming standards, labels, and names
- Determine the operational and visual standards
- Decide whether sound and/or other media will be used
- Determine how and when reviews, tests, and evaluations of the materials are to be conducted

Phase IIIR. The Design Review--The purpose of the design review is to examine the activities performed during the design phase to ensure that: 1)they were performed, 2) they are accurate, and 3) they are complete. The items to be reviewed are the same as those described above. Of special importance is the careful scrutiny of the proposed events of instruction, time allotments, sequences, and interactive procedures with regard to accuracy and relevance to the population and instructional environment.

Phase IV. The Development Phase--The purpose of this phase is to create the computer-based materials according to the design, specifications, and guidelines previously established. The end results are the materials as close to final as possible so that they can be tested by members of the target population. There is no formal review for this phase other than Phase V. However, you may want your manager to establish mini-reviews based upon completion of logical segments of the materials.

Phase V. Test/Evaluation Phase--This phase ensures that the materials meet the instructional goals and are ready to be distributed for use by the target population. This phase should last as long as it takes to determine that there are no longer any problems in the content or operation of the materials. The materials should be thoroughly tested or evaluated according to a previously established Test Plan.

Phase VI. Distribution Phase--The purpose of this phase is to prepare the final materials for distribution and to distribute them. Generally, an encrypted copy of the magnetic media and camera-ready copies of the printed matter are prepared for the publisher. The publisher will then decide how and when it will distribute your materials.

Phase VII. Support Phase--Once your materials have been distributed, people using your courseware will begin to send inquiries, comments on its use, and notices of content or operational errors. No matter how well the courseware is tested, very few products are error-free when distributed.

Summary Comments on the Phases--The phases suggested here are very similar to those used in the development of any software product; they should help organize and schedule the work. While the last two phases may not seem significant, early decisions about how these activities will be performed are necessary because these decisions will make the work involved much easier in the long run. At each phase you need to ensure that the department or group has completed that phase before moving on to the next phase.

Motivating

Normally in a small group motivation is not a management issue; however, in this case the team will consist of people with disparate skills which may cause some internal conflict and require greater attention. Suggested ways for keeping people motivated are:

- Make sure that all communications channels are open
- Involve group members in the decisions to be made
- Make sure that each team member clearly understands his/her responsibilities and potential rewards
- Encourage the team to work together and use the other person's expertise whenever possible

CONCLUSIONS

Project management is a task necessary for the efficient creation of effective computer-based instruction. It involves understanding management concepts, using management techniques, and a management process. The most important concept to remember is that this task, like any other, involves making trade-off decisions.

APTITUDES AND ATTITUDES ABOUT COMPUTERS: TWO STATES COMPARED

Paul J. Will

State University of New York
College at Oswego

ABSTRACT

Responses of 972 Michigan college students, surveyed in 1983-84, and 917 New York students, questioned in 1985-86, form the basis for the study. Demographic information made up half of the 25-item questionnaire with questions about the computer's place in education and specific applications completing the survey. Widespread realization of the importance of computer technology but an uneasiness about its effects is revealed.

Introduction

Almost no studies on aptitude and attitudes of general education students in college computer literacy courses exist and no real attempt to link the two factors has been done. This study was designed to collect data on such information as well as contrast two separate groups, one in Michigan and the other in New York.

The survey instrument, a 25-item questionnaire, was pre-tested and then administered to 972 students during the 1983-84 academic year in all "CPS 100 - Computers and Society" classes at Central Michigan University, Mt. Pleasant, Michigan. The document was limited to one page with the descriptive information about the subjects asked on one side and all attitudinal questions asked on the reverse side.

The questionnaire was given out before any instruction was conducted on the first day of class, preceding even the distribution of the course syllabus. Thus, the stated objectives and course content themselves had no impact on the students' responses. Any form that had no attitudinal responses at all was eliminated from the sample.

Subsequently, a slightly revised version of the same questionnaire was given to all students enrolled in "CS 101 - Survey of Computers and Their Applications," a similar general education course at the State University of New York College at Oswego in the 1985-86 academic year. The same procedure was followed in administering the questionnaire. A total of 917 usable responses was obtained. In both cases, the students were all enrolled in an introductory computer course which fulfills general education curricular requirements at their respective schools. These courses are not conceived as true programming classes, rather their goal is to expose students to computers and their uses.

Aptitudes

The basic demographic information on the two groups is essentially the same with a few notable exceptions. Not only in age but in official classification, the vast majority of the subject groups were underclassmen. The New York sample contained more non-traditional, older students. Gender classification showed a near even split with slightly more females. A majority of the respondents had at least one parent who had attended college. The majors selected by the two samples revealed almost an equal number majoring in business. In the case of New York, the course was actually a requirement for a major in Business Administration whereas in Michigan a second computer course was required with this particular class meeting only general education requirements.

Both the New York and Michigan samples experienced the usual drop between high school and college grade point averages. The data reflects this pattern, e.g., 60-67% of all students had high school GPAs of 3.00 - 4.00 whereas 25 - 34% of them had similar college GPAs. There is also a difference in the grading standards between the two states with Michigan's secondary and 4-year schools appearing to be less rigorous in their evaluation of students.

Another area of contrast was the amount of mathematical preparation students brought into such a course. The popular belief that a math background is necessary for computer learning was not reflected in the survey results. Students in the New York sample had more math training. In both cases, over a third of the students had only an elementary or no math background.

In regard to computer education there is variation between the groups due to curricular differences. As noted before, business students at Central Michigan are required to take another course on computers and the school offers both a major and minor. The State University of New York College at Oswego, on the other hand, requires this particular class for their business majors and does not offer a minor in computer science. Thus, 42% of New York students as contrasted with 20% of Michigan students intend to take only this single course. A higher percentage of students in New York (35% vs. 25%) plan to decide later what to do about additional computer education.

Previous exposure to computers also differs. This may be a reflection of the emerging role of computer education on the secondary level. Twice as many students in New York had a computer course in high school as contrasted with those in Michigan. But there is a two year hiatus between the subject polling, which was a period of rapid expansion for computer literacy courses in high school. A smaller number of students in New York had taken another college computer class.

The extent of computer application outside of school by the respondents was interesting. About one out of four had used computers at home for more than playing video games. Almost half had used computers in a school or job situation. In the New York survey these two catagories were actually separated to obtain more precise data about whether previous usage had occurred in school or in a job situation, but overall the results were parallel.

Attitudes

The attitudinal section of the survey instrument revealed a highly positive attitude toward computers and computer applications in general. One area of striking disagreement between the two groups of respondents was over being uneasy, perhaps fearful, about working with computers. Nearly 43% of the New York students felt this way, but only 22% of the Michigan students were uneasy. The latter response came from a sample group where approximately half had no previous serious exposure to computers in the school, work place, or home. This percentage might be influenced, in part, by the fact that the Michigan course used microcomputers to run application software packages with no significant programming being done in the class.

The New York responses to this question are harder to explain. Almost twice as many students are uneasy, perhaps fearful. Yet, this group as a whole has far more experience with computers. This may suggest that the previous exposure was negative and left the students with new anxiety. In addition, the course at Oswego involves the use of a mainframe to complete a series of programming assignments. Still the difference between the two sets of responses to this particular question is striking.

On all other questions, the groups are in essential agreement only differing in the degree of support. Overall, the Michigan sample is more positive than its New York counterpart. While students in both groups see the importance of computer skills and favor requiring them as part of one's education, they do not believe that they are as basic as reading and writing. Less than a majority saw computer skills as basic skills, but three-fourths would make computer knowledge mandatory for high school graduates and almost the same number would make them a college general education requirement. This is influenced, no doubt, by the overwhelming perception that computer skills will be needed for future employment. Interestingly, far fewer felt that computer applications would be part of their next job. Still the conviction displayed by the respondents about the desirability, almost necessity, of computer knowledge is striking.

In terms of specific computer applications, one finds again a positive attitude. A majority, but certainly not all, would feel comfortable being monitored by a computer in a hospital. The growing involvement of computers in education receives support with a clear majority of students feeling comfortable with the classroom use of computers as teaching aids. The long use of computers in business may reflect the strong approval of the use of business computers to bill accounts and deal with customers. These positive attitudes may be linked to the conviction of the respondents that computers will raise the standard of living for the American people.

A more serious note of caution is sounded amongst all these positive accolades. A clear majority in both samples feel that the use of computers must be sharply restricted in the future, if individual privacy is to be protected. This survey result mirrors other studies of the general public done in the past and, in this case, is particularly significant because the question was asked before any course content, which includes the dangers of computer surveillance, was presented.

Thus, looking at the two groups one finds general agreement about the desirability of computer knowledge and many computer applications. Yet, beneath this rosy surface, there lie some concerns about working with computers themselves and their possible misuse. The media blitz about the Computer Revolution has had its effect: young Americans realize that the computer is a machine that is presently affecting them and will continue to impact their lives.

TABLES

I. Aptitude Responses

Question		New York		Michigan	
My age catagory is:	17 - 19	514	56.1%	584	60.1%
	20 - 21	265	28.9%	297	30.6%
	22 - 25	81	8.8%	61	6.3%
	26 - 29	16	1.8%	15	1.5%
	30 or over	40	4.4%	15	1.5%
My college classification is:					
Freshman		303	33.0%	242	24.9%
Sophomore		318	34.7%	470	48.4%
Junior		186	20.3%	165	17.0%
Senior		89	9.7%	92	9.5%
Graduate or Special		20	2.2%	2	.2%
My sex classification is:	Female	478	52.1%	498	51.4%
	Male	432	47.1%	470	48.5%
At least one of my parents attended college	Yes	597	65.1%	586	60.3%
	No	317	34.6%	386	39.7%

COMPUTER APTITUDES AND ATTITUDES

Question	New York		Michigan	
My college major is:				
Business	307	33.5%	348	35.8%
Education	48	5.2%	NA	
Liberal Arts - non-science/non-math	193	21.1%	89	9.2%
Liberal Arts - science/math	81	8.8%	116	11.9%
Fine and Applied Arts	NA		146	15.0%
another area or undeclared at present	287	31.3%	273	28.1%
My high school grade point average was:				
3.50 - 4.00	125	13.6%	231	23.8%
3.00 - 3.49	428	46.7%	418	43.1%
2.50 - 2.99	227	24.8%	260	26.8%
2.00 - 2.49	46	5.0%	47	4.8%
unknown	90	9.8%	14	1.4%
My college grade point average is:				
3.50 - 4.00	49	5.3%	78	8.0%
3.00 - 3.49	180	19.6%	255	26.3%
2.50 - 2.99	269	29.3%	343	35.4%
2.00 - 2.49	251	27.4%	141	14.5%
none at this time	165	18.0%	113	11.6%
under 2.00	NA		40	4.1%
The level of college math I have completed is:				
calculus or higher	264	28.8%	156	16.1%
one or more courses in algebra	288	31.4%	218	22.5%
basic/general math	194	21.2%	183	18.9%
business math	49	5.3%	129	13.3%
none	121	13.2%	283	29.2%

WILL

Question		New York		Michigan	
I intend to:					
take only this course on computers		383	41.3%	197	20.3%
take at least one more course on computers		188	20.5%	314	32.3%
take a major in computer science or information science (management information systems)		21	2.3%	121	12.5%
take a minor in computer science or business data processing		NA		99	10.2%
decide later what I will do		318	34.7%	239	24.6%
I took at least one computer course in high school	Yes	478	52.1%	237	24.4%
	No	438	47.8%	735	75.6%
I have had at least one previous college computer course	Yes	100	10.9%	226	23.3%
	No	815	88.9%	745	76.7%
I have used computers in school or in a job situation	Yes	NA		481	49.5%
	No			491	50.5%
I have used computers in a job situation	Yes	226	24.7%	NA	
	No	688	75.0%		
I have used computers in school (but not a formal course)	Yes	419	45.7%	NA	
	No	496	54.1%		
I have used computers at home for more than playing video games	Yes	230	25.1%	197	20.3%
	No	684	74.6%	775	79.7%

COMPUTER APTITUDES AND ATTITUDES

II. Attitudinal Responses

Question		New York		Michigan	
I am uneasy, perhaps fearful, about working with computers	Yes	393	42.9%	215	22.5%
	No	523	57.0%	741	77.5%
Computer skills are as basic as reading and writing skills	Yes	360	39.3%	421	44.8%
	No	553	60.3%	517	55.1%
Basic knowledge of computers should be mandatory for graduation from high school	Yes	695	75.8%	731	76.5%
	No	218	23.8%	224	23.5%
Computer literacy should be a general education requirement for all college students	Yes	645	70.3%	742	78.1%
	No	267	29.1%	208	21.9%
Computer skills will be needed for employment in the future	Yes	795	86.7%	873	92.0%
	No	116	12.7%	76	8.0%
Computer applications will be part of my next job	Yes	370	40.4%	478	51.5%
	No	537	58.6%	448	48.3%
If I were hospitalized, I would feel comfortable being monitored by a computer	Yes	455	49.6%	589	62.6%
	No	456	49.7%	352	37.4%
I would feel comfortable in a classroom situation where the computer is used as a teaching aid	Yes	548	59.8%	607	64.2%
	No	363	39.6%	338	35.8%
I would feel comfortable with a business that used computers to bill their accounts and deal with their customers	Yes	667	72.7%	782	82.5%
	No	245	26.7%	166	17.5%
Computers will raise the standard of living for the American people	Yes	597	65.1%	700	75.6%
	No	306	33.4%	224	24.2%
If individual privacy is to be protected, the use of computers must be sharply restricted in the future	Yes	503	54.9%	527	56.7%
	No	374	40.8%	399	42.9%

ADCIS Proceedings, November 1986

STRUCTURED MUSIC CAI PROGRAMMING AND DATA STRUCTURES

Ann K. Blombach

The Ohio State University

Abstract

Using UCSD MacAdvantage Pascal and a pitch representation system originally developed by Walter Hewlett (Center for Computer Assisted Research in the Humanities, Menlo Park, CA), we are programming "semi-intelligent" music aural training courseware for the Macintosh, based on our earlier GAMUT (Graded Aural MUsic Training) courseware. Using Ohio State's mainframe computer capabilities to good advantage, we had previously developed highly sophisticated exercises and feedback, as well as provisions for keeping almost unlimited records of each student's work. The microcomputer music CAI system we are currently developing makes it possible to maintain the integrity of our pedagogical concepts while permitting a student to enter his answers--without using the typewriter keyboard or any sort of piano keyboard--in a manner more nearly analogous to the traditional music notation and reading skills every music student must learn. Hewlett's pitch representation system and my algorithms based on that system have made it possible to translate quickly and easily among the diverse forms of musical data required for exercise generation, answer entry, graphic display, sound generation, and answer evaluation, resulting in a savings of both calculation time and memory usage which has been a critical factor in our microcomputer programming success. In addition, relying heavily on structured programming concepts, we are, in effect, producing a higher level music programming language by building a library of multi-use program components which have application far beyond the immediate task.

Background

In September, 1980, The Ohio State University began developing computer-assisted courseware to meet the needs of the 350 music students enrolled in our aural training courses. The programs, written in the Phoenix computer language, were intended to provide each student with individualized drill in all aspects of aural training. Magnavox Orion terminals with plasma (touch-sensitive) screens were linked to an Amdahl mainframe computer via a 1200-baud direct phone line and interfaced to four-voice "music boxes" (sound generators) designed and built by a Columbus-based company. The competency-based programs included modules on scales, melodic and harmonic intervals, duration patterns, melodies, triads, seventh chords, and harmonic dictation.

The link to the Amdahl made it possible to provide exercises and feedback that were quite sophisticated, as well as allowing us to keep almost unlimited records of each student's work. The records kept for each student could be used for evaluation of the student's progress and diagnosis of the student's problems to provide maximum individualization of instruction. The system also provided extensive opportunities for research into the acquisition of aural skills.

Unfortunately, the system suffered from a number of problems whose solutions were beyond our control. For example, our equipment was obsolete before our first programs were operating (Magnavox stopped producing Orions soon after we bought our first one); because of our unique equipment set-up, the courseware was not transportable; and the students were continually annoyed by the slowness of the time-sharing system on the mainframe computer. Some of the most frustrating problems, however, were entirely our own fault--a fact which made them no easier to solve. The original programs had been written primarily by one programmer who, at my urging, tried to do as much programming as possible in as short a time as possible. Because of our haste, there was very little documentation explaining how the programs worked, and we were constantly plagued by apparently untraceable program errors. In addition, because we were learning about program design as we worked, the original programs were poorly structured and the programming procedures were not at all consistent. In short, we had developed a conceptually sophisticated pedagogical system which both we and our students all too often found unbelievably frustrating to use.

By September 1984 we had decided to switch to microcomputers. Our intention was to implement the pedagogical concepts of our existing courseware on a microcomputer, making as few compromises as possible. After several months of careful consideration and investigation, I chose the Macintosh 512K, with an extra disk drive for use during program development.

Structured Programming

In selecting a programming language for the project, my first requirement was that the language have the capacity to maintain the sophistication of our original mainframe programs. A factor almost as important, however, was the lesson we had learned from our previous painful "unstructured" programming experiences: We needed a structured language which would help us in designing and maintaining our programs. I chose UCSD MacAdvantage Pascal. Although Pascal at first seemed unnecessarily strict and confining (a perception which was a good indication of how much we needed the structure the language provided), we soon began to benefit from those very Pascal restrictions which force the programmer to organize programs carefully and thoroughly. One additional advantage of choosing Pascal was that the Macintosh "Toolbox" routines --the graphs, sound, and other routines resident in the Macintosh ROM--were written in Pascal. Because UCSD Pascal and the Lisa Pascal in which the Toolbox routines were written are not entirely compatible, the explanations in *Macintosh Revealed* and *Inside Macintosh* are not always directly applicable to a UCSD application. On the whole, however, the advantages of the similarities between the languages have outweighed the disadvantages of their differences.

We began by breaking each clearly separable programming task into its component parts, preparing successively more detailed outlines consisting of "pseudo-code" statements. Initially, we ignored most "branching" kinds of decision statements, trying instead to focus on the basic elements which were necessary to carry out the task at hand in the most straightforward way. For example, our first draft of a section of pseudo-code designed to set up a single example for a melodic interval recognition program included the following statements:

```
Randomly choose the interval
Randomly choose the clef
Randomly choose the direction
Randomly choose the beginning pitch
Draw the screen
```

We then split each of these statements into its component parts, if any, continuing the process until we arrived at the most fundamental elements of the task. "Randomly choose the interval" is, in itself, a relatively simple problem, although we might want to list specific interval choices or numbers of intervals to choose from, depending on the level of difficulty. "Randomly choose the clef" and "Randomly choose the direction" cannot be divided into any sub-tasks, but the particular clefs available for choice must be specified. "Randomly choose the beginning pitch" might seem to be equally straightforward but, in fact, this statement represents a somewhat more complex problem. Given that the interval, clef, and direction have already been chosen, the beginning pitches which are acceptable choices will necessarily be limited to a specific subset of all possible pitches. For example, if the chosen interval were a descending major seventh on the treble clef, we would probably not want the beginning pitch to be middle C. The statement, "Draw the screen," represents a much different kind of problem than the preceding statements. In our initial pseudo-code draft, however, we included "Draw the screen" as a placeholder for future programming tasks.

A more detailed version of pseudo-code, with some explanatory and admonitory material inserted, included these statements:

```
Randomly choose the interval
  *For level 1, randomly choose the
    answer from 4 randomly chosen
    interval choices
  For level 2, randomly choose the
    answer from 6 randomly chosen
    interval choices
  For level 3, randomly choose the
    answer from all possible intervals*
Randomly choose the clef
  *Choose from treble, bass, alto, and
    tenor clefs*
Randomly choose the direction
  *Up or down, allowing also for "same,"
    in the case of a unison*
Randomly choose the beginning pitch
  *Check to make sure that both the
    beginning pitch and the second pitch
    are within acceptable range for clef,
    allowing 2 leger lines above and
    below the clef
  If the beginning pitch does not work,
    choose another one and repeat the
    check*
Draw the generic screen
  *program name, level info, staff,
    answer boxes, "accidental" boxes,
    others?*
Draw the information specific to this
exercise on the screen
  *clef, beginning pitch, answer choices*
```

The next step was the translation of the basic components into Pascal functions and procedures which could be used throughout the entire series of programs. In other words, in programming even these simple tasks, the objective was to make each function and procedure as universally applicable as possible. For example, the need for randomly choosing the clef for presentation of an exercise is by no means limited to interval drills; such a function would also be useful in several other modules. Similarly, checking to see whether a particular pitch is in an acceptable range for a particular clef is always a necessary step in the pitch notation process, regardless of the subject matter of the specific exercise. As we develop each program, then, we are also building a library of multi-use program components which should have application far beyond the immediate problem.

The procedures and functions called in the section of program code which represents the first four statements of pseudo-code listed above are almost as straightforward as the pseudo-code (for clarity, function and procedures arguments have been omitted):

```
Choose_Interval_Choices
Right_Answer:=Choose_Answer *from the
    interval choices*
Clef:=Choose_Clef
Direction:=Choose_Direction
Repeat
  Pitch_1:=Choose_Beginning_Pitch
  Pitch_2:=Calculate_Second_Pitch
Until Range_OK *for both pitches*
```

In the final version of the program, some additional refinements have been added (for example, it allows for the two different notationally correct answers for the second pitch of a tritone interval), but--except, of course, for the function and procedure arguments which must be included--this brief amount of code is essentially a working music CAI program.

One unexpected side benefit of this programming method has been our discovery that the only realistic way to keep everyone up to date on what has already been accomplished is to keep the documentation current. Documentation concerning exactly how each procedure and function operates along with an explanation of the arguments which are required must be completed in a timely fashion so that all the programmers involved in the project can benefit from all the work others have already completed. In effect, as we develop our programs, we are producing a higher level music CAI programming language, written in UCSD Pascal.

Data Structures

The basic pitch data structure we are using was derived from Walter Hewlett's (Center for Computer Assisted Research in the Humanities, Menlo Park, CA) system of pitch representation. For each pitch name (letter name plus accidental, ranging from double flat to double sharp), his system provides a unique pitch representation number which distinguishes among enharmonic spellings of the same pitch (see Table 1).

For purposes of interval identification, these Hewlett Representation numbers (HR numbers) are easy to use: subtracting one number from another results in a unique interval identification number which also preserves the differences in enharmonic spellings (see Table 2).

We allow for representation of the octave register (using American standard octave register designation, where middle C is C4) by multiplying the octave register by 40 and adding the pitch's HR number. For example, C natural 4 = (4* 40) +2, or 162; D double flat 4 = (4* 20) +6, or 166.

TABLE 1: Hewlett system of pitch representation

d = double flat
b = flat
n = natural
s = sharp
x = double sharp

Cd	0	Fs	20
Cb	1	Fx	21
Cn	2	--	22
Cs	3	Gd	23
Cx	4	Gb	24
--	5	Gn	25
Dd	6	Gs	26
Db	7	Gx	27
Dn	8	--	28
Ds	9	Ad	29
Dx	10	Ab	30
--	11	An	31
Ed	12	As	32
Eb	13	Ax	33
En	14	--	34
Es	15	Bd	35
Ex	16	Bb	36
Fd	17	Bn	37
Fb	18	Bs	38
Fn	19	Bx	39

TABLE 2: Hewlett Interval Identification Numbers

0	P1	23	P5
1	A1	24	A5
4	d2	27	d6
5	m2	28	m6
6	M2	29	M6
7	A2	30	A6
10	d3	33	d7
11	m3	34	m7
12	M3	35	M7
13	A3	36	A7
16	d4	39	d8
17	P4	40	P8
18	A4	41	A8
22	d5		

In addition to providing a simple method for interval calculations, the system lends itself to the design of truly elegant algorithms which facilitate the various musical data transformations necessary for a broad range of computer music programming tasks, including exercise generation, answer entry, graphic display, sound generation, and answer evaluation. In an interactive learning situation which requires rather complex graphics displays and "mouse" tracking along with sound generation routines and seemingly instantaneous computer reactions to almost continuous student inputs, the speed of calculation is a vital element in the courseware's success. The efficiency of data storage and musical manipulation procedures is a critical factor in the speed of the computer's response. I am happy to report that the implementation of Hewlett's system has made the computer processing time practically transparent to the student. In other words, the system works.

Pitch data, including letter names, accidental symbol font characters, and assorted "base" numbers necessary for calculating various numerical pitch and graphic music notation information, are stored in arrays of records. (In Pascal, the record structure provides for data of different types to be stored as a unit; storing the records in arrays allows specific information to be accessed directly and easily.) The interval name and HR interval identification number are also stored in an array of records.

Examples 1 and 2 show the contents of the arrays of pitch records and accidental records respectively. All the basic information needed to perform any necessary pitch "translations" is stored in these two arrays. The algorithms used to reference these arrays and the interval identification array are very straightforward.

EXAMPLE 1: Contents of pit_id array

index	let	pcNum	hrBase
0	'C'	0	0
1	'D'	2	6
2	'E'	4	12
3	'F'	5	17
4	'G'	7	23
5	'A'	9	29
6	'B'	11	35

EXAMPLE 2: Contents of acc_id array

index	acc	aNum
0	'd'	-2
1	'b'	-1
2	'n'	0
3	's'	1
4	'x'	2

Example 3 is a procedure that establishes the specific array references for a given HR number, setting the values for variables which are then used in several other procedures and functions to calculate various equivalencies.

EXAMPLE 3: Hr_To_Pit Procedure

```
Procedure Hr_To_Pit
    (var hrNum, pitNum, orNum:  Integer;
    var badNote:  Boolean);

  var
    letNum:  Integer;

  begin
    orNum:= hrNum div 40;
    letNum:= hrNum mod 40;
    if letNum = 17
      then letNum:= letNum + 1;
    accNum:= letNum mod 6;
    if accNum = 5
      then badNote:= true
      else begin
        pitNum:= letNum div 6;
        badNote:= false;
      end;
  end (*Hr_To_Pit*);
```

If, for example, hrNum = 162, then this procedure sets the pit_id index (pitNum) to 0, the acc_id index (accNum) to 2, and orNum = 4--i.e., hrNum 162 = CN4.

Given the HR number, Hr_To_Pit calculates pitNum and accNum for use in referencing the pit_id (Example 1) and acc_id (Example 2) arrays respectively. It also calculates the octave register number (orNum). If the HR number does not represent a valid pitch in the system, the Boolean variable badNote is set to true.

Examples 4 and 5 illustrate how these array references are used to derive necessary pitch information in subsequent procedures and functions.

EXAMPLE 4: Calc_PC_Num Function

```
Function Calc_PC_Num (pitNum, accNum,
    orNum:  Integer):  Integer;

  begin
    Calc_PC_Num:= (orNum * 12)
      + pitNum + accNum;
  end (*Calc_PC_Num*);
```

EXAMPLE 5: Pit_To_Hr Function

```
Function Pit_To_Hr (hrBase, accNum,
    orNum:  Integer): Integer;

  begin
    Pit_To_Hr:= (orNum * 40) + hrBase
      + accNum;
  end (*Pit_To_Hr*);
```

The Calc_PC_Num function calculates the pitch class number for a given HR number, using the array references established by the Hr_To_Pit procedure. The Pit_To_Hr function returns the HR number of a given pitch. This function is an important part of the answer checking routines since it allows us to identify an incorrectly notated interval quickly, thereby providing the basis for immediate, intelligent feedback (for example, "You have correctly identified the interval as m6 but you have notated M6. Would you like to hear these intervals?").

Thanks to the capabilities of the Macintosh, the structured programming techniques required in UCSD Pascal, and Hewlett's straightforward music data structure, we have been able to make a very significant amount of progress in a very short time. The benefits of learning from our past mistakes have certainly contributed to our current successes as well. We are well on our way to developing a complete series of aural training lessons on the Macintosh and, at the same time, a music CAI development system that--we hope--can eventually be made available to other music CAI programmers.

Contact: Ann K. Blombach
School of Music
The Ohio State University
1866 College Road
Columbus, OH 43210

ADCIS Proceedings, November 1986

COMPUTER-BASED RECOGNITION OF PERCEPTUAL PATTERNS AND LEARNING STYLES IN MELODIC DICTATION EXERCISES

Fred T. Hofstetter

University of Delaware

ABSTRACT

During the 1981-82 academic year, nineteen freshman music majors participated in an experiment in which their response data were saved as they worked through 18 units of melodic dictation exercises in the University of Delaware's GUIDO(r) system. Analysis of the student data base led to the identification of five general principles of melodic perception. First, when students miss a melodic leap, they tend to underestimate its size. Second, most wrong answers lie within a diatonic step of the stimulus. Third, implied harmonics affect the perception of large intervals. Fourth, notational problems affect the answering of exercises in the minor mode, where TI and TE are confused with each other, as are LA and LE. Fifth, the scale degree upon which an interval is built can significantly affect the likelihood that a student will answer it correctly.

Analysis of individual student learning styles showed that as the level of difficulty went up, students tended to increase instead of decrease the speed of dictation. On the average, students asked for each melody to be dictated twice. Neither speed of dictation, mode of the exercise (major or minor), nor the number of accidentals in the key signature correlated with achievement. Although students took less time to answer with solfeggio syllables than with pitches, neither solfeggio syllables nor pitches correlated with achievement. The tonic pitch was answered correctly most often (93%), whereas the leading tone had the lowest score (85%).

Techniques of exploratory data analysis were employed to determine the effect of specific intervals upon student achievement. A box plot revealed an overall pattern in which achievement tended to decrease as intervallic size increased, except for fifths. A sharp rise in performance was noted for ascending fifths, whereas a corresponding drop occurred for descending fifths. A median polish was performed on a table of data containing the percent correct for each student on each interval. The results revealed a group of five students who performed particularly well on ascending fifths, but poorly on descending fifths. Detailed analysis of the confusion patterns led to the discovery of five errors that occur during student perception of melodic leaps.

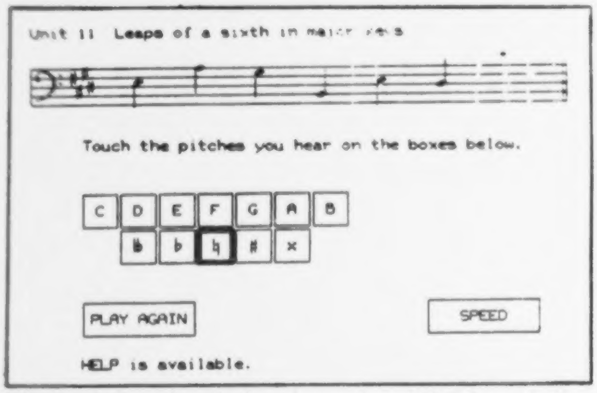
INTRODUCTION

This is the fifth in a series of articles reporting the results of experiments in aural training using the University of Delaware's GUIDO system. The first four articles (Hofstetter 1978, 1979, 1980, 1981) dealt with student acquisition of dictation skills in harmonics, intervals, chord qualities, and rhythms; this article deals with melodic dictation.

It is hoped that as knowledge accumulates through this process of experimentation, a cognitive model of how students learn to internalize and conceptualize the music they hear can be developed. The present study of student perception of melodic dictation exercises is one step in this process.

APPARATUS

All of the melodic dictation exercises used as stimuli in this experiment were presented to students by means of the melodic dictation program in the University of Delaware's GUIDO system. Figure 1 shows a sample screen display from this program. At the top of the screen, students are shown the number



Unit 11 Leaps of a sixth in major keys

Touch the pitches you hear on the boxes below.

HELP is available.

Figure 1--Sample display from the GUIDO melodic dictation program by Fred T. Hofstetter and William H. Lynch. Copyright (c) 1981 by the University of Delaware.

and title of the unit in which they are working, and then GUIDO displays the notes that the students have answered correctly so far. All of the stimuli are presented as quarter notes, and students are not required to enter any rhythmic values. Students can answer the exercises sequentially from left to right, or they can use arrow keys (<-- and -->) to position a pointer over the notes they wish to answer.

When a melody is dictated, the students are told to "Touch the pitches you hear on the boxes below." The boxes shown in Figure 1 are called pitch boxes. First, the student presses the accidental (if there is one), followed by the letter name. If the student gets an accidental wrong, GUIDO prints an arrow beneath the box that was falsely touched; this arrow points in the direction of the correct accidental. Similarly, if the student misses a letter name, an arrow pointing in the direction of the correct answer is printed above the wrong letter. Students must work on an exercise until they get all of the notes correct. An answer key called ANS is activated after a mistake. Students can press the ANS key to see the correct answer to the note they missed. ANS does not function for notes that have not been attempted.

Toward the bottom of the screen are two large boxes marked PLAY AGAIN and SPEED. Students may ask for an exercise to be played again by pressing the PLAY AGAIN box. When students press the SPEED box, a row of boxes containing a selection of speeds from 30 to 300 quarter notes per minute is displayed, and the students are asked to touch the box that contains the speed they want.

Some of the exercises used in this study required students to answer with solfeggio syllables instead of pitches. Figure 2 shows the touch boxes that replace the pitch boxes when solfeggio syllables are used.

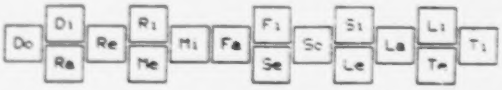


Figure 2--Touch-sensitive solfeggio syllable boxes used in the GUIDO melodic dictation program. Copyright (c) 1981 by the University of Delaware.

As with all of the programs in the GUIDO system, dictation exercises are generated according to a master table in which a curriculum has been specified. Table 1 is a master table that shows the 18 melody units used in this experiment. Columns one and two show the number and title of each unit. Column three shows whether pitches or solfeggio syllables appeared on the screen. Column four contains the number of sharps or flats allowed in the key signature. This number was used by an automatic syntactic question generator (Hofstetter, 1985) to randomly select a key for each exercise. Column five indicates whether the exercises were presented in major or minor mode. When solfeggio syllables were used, the tonic was defined as DO in both major and minor modes. Columns six and seven show the number of notes contained in the shortest and longest exercises generated in each unit.

At the bottom of Table 1 are listed four variables that can be manipulated in the melody program, but which had the same value for each of the eighteen units in this experiment. As was seen in Figure 1, students

TABLE 1
Master table showing the 18 melody units used in the experiment.

Unit	Title	Answer Display	Accidentals Allowed	Mode	Shortest Melody	Longest Melody
1	Short Scalewise Melodies in Major Keys	Pitch	2	Major	7	7
2	Longer Scalewise Melodies in Major Keys	Pitch	3	Major	7	9
3	Longer Scalewise Melodies: Solfeggios	Solfeggio	7	Major	8	10
4	Leaps from the Tonic Triad in Major Keys	Pitch	3	Major	7	8
5	Melodies with Leaps from I and V in Major Keys	Pitch	3	Major	6	8
6	Leaps from I and V: Solfeggios	Solfeggio	7	Major	6	8
7	Leaps from I, IV, V, and VII in Major Keys	Pitch	4	Major	8	10
8	Short Melodies in Minor Keys	Pitch	4	Minor	7	8
9	Longer Melodies in Minor Keys	Pitch	4	Minor	9	10
10	Longer Melodies in Minor Keys: Solfeggios	Solfeggio	4	Minor	9	10
11	Leaps of a Sixth in Major Keys	Pitch	4	Major	6	7
12	Leaps of a Sixth in Minor Keys	Pitch	4	Minor	6	7
13	Leaps of a Seventh in Major Keys	Pitch	4	Major	6	10
14	Leaps of a Seventh in Minor Keys	Pitch	4	Minor	6	9
15	Sixths and Sevenths in Major Keys: Solfeggios	Solfeggio	4	Major	5	10
16	Sixths and Sevenths in Minor Keys: Solfeggios	Solfeggio	7	Minor	5	10
17	More Angular Melodies in Major Keys	Pitch	4	Major	6	10
18	More Angular Melodies in Minor Keys	Pitch	4	Minor	7	10

Note: Variables that can be manipulated in the melody program, but which had the same value for each of the 18 units in this experiment:

Speed: Student Choice
Time Limit: 10 Seconds Per Note
Number of Plays: Student Choice
Advancement Criterion: 4 out of 5 at 75% Correct

were given the choice of replaying exercises as often as they wanted and of changing the speed of dictation. The time limit was set at 10 seconds per note, and if students did not answer within that amount of time, the exercise was judged incorrect. The advancement criterion was four out of five at 75% correct; this means that an exercise was considered to be correct if a student correctly answered 75% of its pitches on the first try. To achieve mastery in a given unit, the students had to answer four out of five exercises at this 75% level.

PROCEDURE

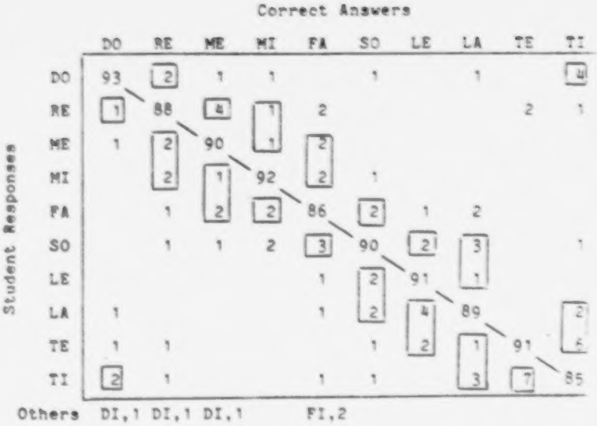
During the 1981-82 academic year, 19 freshman music majors completed 4,180 melodic dictation exercises in partial fulfillment of the requirements for their freshman ear-training course. For each exercise, the computer saved the name of the student, the number of the unit, the complete stimulus, the student's first response to each pitch in the exercise, the key signature, whether the exercise was judged correct, the number of seconds it took the student to get the right answer, the number of times the student played the exercise, the speed of dictation, the mode (major or minor) of the exercise, and the answer format (pitches or solfeggio syllables). When the students answered a pitch incorrectly, they were required to keep trying until they got it right. In this study, however, only the first response to each pitch was considered for analysis.

Five kinds of analyses were applied to the student response data base. First, cross tabulations of correct answers and student responses were made to determine whether there were any confusion patterns in the student perception of melody. Second, a box plot was displayed to permit visual comparison of the effect of intervals on student achievement. Third, a median polish procedure was used to explain an unexpected pattern found in a graph of mean scores for each interval. Fourth, second-order cross tabulations were made in order to describe the effect of scalar position upon the perception of intervals in a melody. Finally, a multiple correlation of instructional variables was made to measure student learning styles. The results of these five analyses are discussed in turn as follows.

RESULTS

Since the student data base contained all exercises completed by each student throughout the curriculum, some of the exercises were done while students were just learning, while others were done as students approached mastery in a given unit. In making the cross tabulations of correct answers and student responses, it was decided to include only those exercises that counted toward mastery--that is, those in which at least 75% of the pitches were answered correctly on the first try. In this way, the cross tabulations show the most persistent confusion tendencies.

Figure 3 shows the cross tabulation of correct answers and student responses for melodic dictation exercises that were judged correct. Each scale degree exhibits a common confusion pattern in that the highest percentage of mistakes occurs due to confusions that lie within one diatonic scale degree of the stimulus. Boxes drawn on the matrix mark such confusions.



Correct Answers

Student Responses

	DO	RE	ME	MI	FA	SO	LE	LA	TE	TI
DO	93	2	1	1		1		1		4
RE	1	88	4	1	2				2	1
ME	1	2	90	1	2					
MI		2	1	92	2	1				
FA		1	2	2	86	2	1	2		
SO		1	1	2	3	90	2	3		1
LE					1	2	91	1		
LA	1				1	2	4	89		2
TE	1	1				1	2	1	91	6
TI	2	1			1	1		3	7	85
Others	DI,1	DI,1	DI,1		FI,2					

Figure 3--Cross tabulation of correct answers and student responses for exercises that were judged correct.

Table 2 contains a rank-order listing of scale degrees according to percent correct. The range is small, from the highest score of 93% for DO, to a low of 85% for TI. The low score can be explained by considering the last column of Figure 3, which shows how students mistook TI for TE 6% of the time due to notational errors made in the minor mode. Similarly, the most common confusions for TE and LE occured with the major-mode counterparts of TI (7%) and LA (4%). Such major-minor mode confusions occurred much less frequently (1%) for MI and ME, which are not subjected to chromatic alterations as melodies ascend and descend the minor scale.

TABLE 2
Rank order list of scale degrees by percent correct.

Syllable	% Correct
DO	93
MI	92
LE	91
TE	91
ME	90
SO	90
LA	89
RE	88
FA	86
TI	85

The second phase of this study sought to determine whether the intervallic relationship between a pitch and its predecessor influences its perception. For each of the nineteen subjects, separate scores were computed for twelve interval categories, namely, ascending seconds, thirds, fourths, fifths, sixths, and sevenths, and their descending counterparts. Only intervals in which students correctly answered the first note on the first attempt were included in this computation. The categories of seconds, thirds, sixths, and sevenths included both major and minor intervals. Fourths and fifths were always perfect. No diminished or augmented intervals were presented as stimuli in this experiment, although they were occasionally found in the responses.

Figure 4 contains a box-and-whisker plot (Tukey, 1977, p. 40) of percents correct for each interval. A plus sign (+) shows the mean score for each interval. The tops and bottoms of the boxes show the upper and lower quartiles, and the whiskers (dashed lines) show the range of the data.

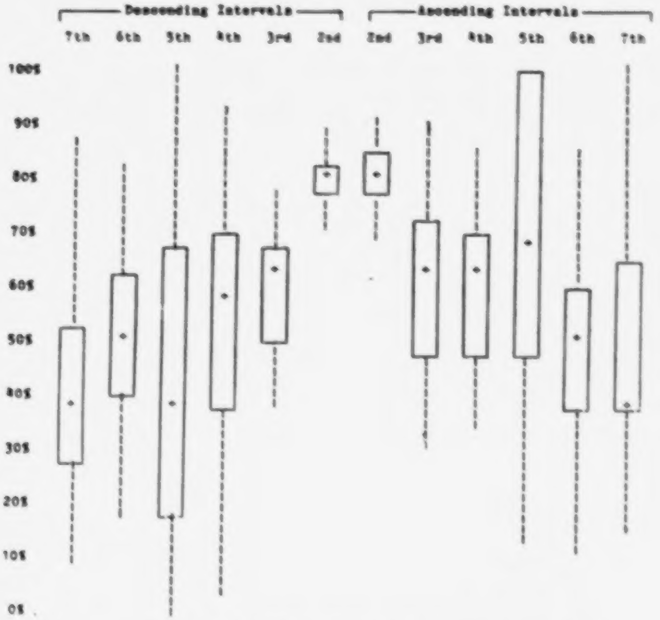


Figure 4--Box-and-whisker plot of percents correct for each interval category.

Two trends are revealed in the box-and-whisker plot. First, as intervals increase in size from seconds to sevenths, the means tend to drop. Students perform better on small intervals than on large ones. Second, as intervals increase in size from seconds to sevenths, the boxes get longer. All students tend to score high on seconds, but as intervallic size increases, so also do the problems students have in answering correctly. One would expect student scores to decrease as intervallic size increases. However, the graph drawn in Figure 5 shows how the means do not decrease evenly as a function of intervallic size. Both on the ascending and descending sides of the graph, it is the interval of a fifth that departs from the expected pattern. On the ascending side, students perform better on fifths than on fourths, causing an unexpected peak in the graph. On the descending side, students perform more poorly on fifths than on sixths, causing a downward dip.

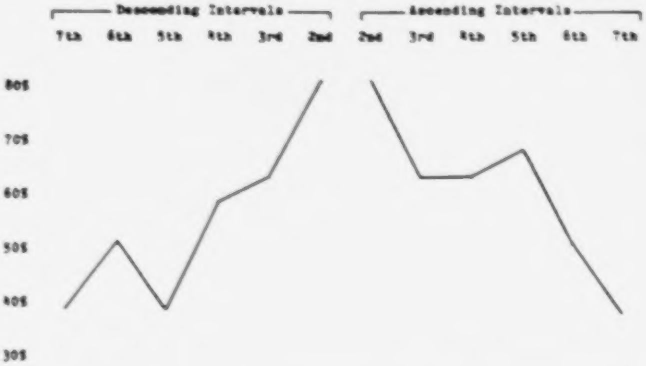


Figure 5--Graph of the means in the box-and-whisker plot.

The third phase of the analysis used a median polish procedure (Tukey, 1977, pp. 363-400) to describe the interactions between descending and ascending fifths. The median polish looked for unusual cases in the pattern of interval scores for each student and resulted in a table of residuals. The residuals reveal data points that cannot be accounted for by row and column effects. Table 3 contains a coded representation of these residuals. The plus signs (+) indicate data points that lie in the upper quartile, and the asterisks (*) probable outliers above the mean. Similarly, the minus signs (-) indicate data points in the lower quartile, and the double minus (=) probable negative outliers. By studying this table, it can be seen how five subjects--numbers 6, 10, 11, 14, and 15--amplify the pattern seen in figure 5 by having remarkably low scores on descending fifths and unusually high scores on

TABLE 3
Coded results of the median polish.

	Descending Intervals						Ascending Intervals					
Students	7th	6th	5th	4th	3rd	2nd	2nd	3rd	4th	5th	6th	7th
1	-	-	+	.	.	.	.	+	.	.	-	+
2	-	.	+	+	.	.	.	+	+	.	-	=
3	-	-	+	*	.	.	.	.	.	-	.	-
4	.	+	.	.	.	.	.	+	.	.	=	-
5	+	+	=	-	.	+	+	.	.	.	.	-
6	-	-	-	+	.	.	.	.	.	+	-	.
7	.	.	.	-	.	.	.	.	.	-	*	*
8	+	.	+	.	.	.	.	-	.	.	-	*
9	+	.	+	+	.	-	-	.	.	.	.	.
10	.	+	-	+	.	-	-	-	.	+	+	.
11	.	+	-	.	+	.	.	+	.	+	=	-
12	-	-	*	+	.	.	.	.	.	-	+	.
13	+	-	+	.	+	-	.	.	-	+	.	-
14	.	-	-	.	+	.	+	.	-	*	.	.
15	-	+	=	+	.	.	.	-	-	*	.	.
16	.	.	*	+	.	.	.	-	+	.	=	.
17	+	.	.	+	.	.	.	.	.	-	.	.
18	+	.	-	-	.	.	.	.	-	=	+	+
19	+	+	-	.	.	+	+	.	.	-	.	-

Key to Coded Data Points:

.	inner quartile
+	upper quartile
-	lower quartile
*	positive outlier
=	negative outlier

ascending fifths. Figure 6 contains a superimposition of response curves that graphically portrays the similarities among the response patterns of these five students.

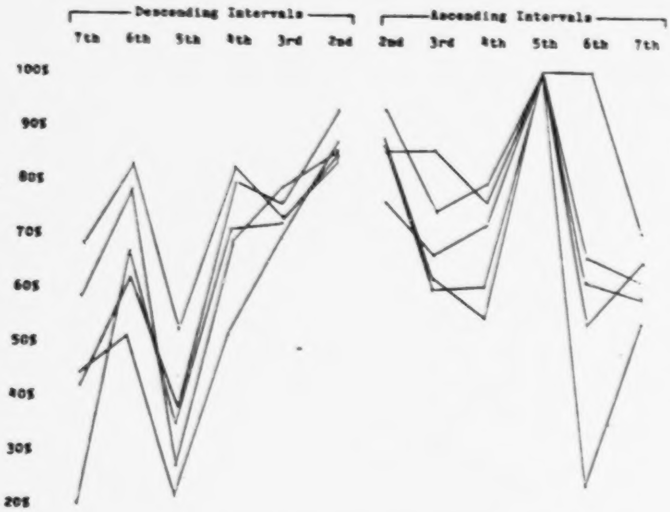


Figure 6--Superimposition of the response curves of subjects 6, 10, 11, 14, and 15

TABLE 4
Primary confusions with ascending intervals.

Interval	Stimulus	% Correct	Primary Confusions*
Minor Seconds	RE-ME	92%	None
	MI-FA	85%	None
	SO-LE	80%	None
	TI-DO	88%	None
Major Seconds	DO-RE	85%	None
	RE-MI	91%	None
	ME-FA	90%	None
	FA-SO	85%	None
	SO-LA	78%	None
	LA-TI	87%	None
Minor Thirds	DO-ME	73%	DO-RE (13%)
	RE-FA	42%	RE-MI (31%), RE-ME (13%)
	MI-SO	78%	MI-FA (11%)
	FA-LE	52%	FA-SO (21%), FA-LA (11%)
	LA-DO	68%	LA-TI (23%)
	TI-RE	64%	TI-DO (24%)
Major Thirds	DO-MI	77%	None
	ME-SO	66%	ME-FA (22%)
	FA-LA	83%	FA-SO (11%)
	SO-TI	61%	SO-TE (15%)
Perfect Fourths	DO-FA	52%	DO-ME (15%), DO-MI (11%)
	RE-SO	46%	RE-FA (37%)
	ME-LE	46%	ME-SO (43%)
	SO-DO	75%	SO-TI (12%)
Perfect Fifths	DO-SO	70%	DO-MI (13%)
	SO-RE	61%	SO-DO (16%)
Minor Sixths	DO-LE	81%	None
	MI-DO	62%	MI-TI (31%)
	SO-ME	59%	SO-RE (18%)
	LA-FA	67%	None
Major Sixths	ME-DO	66%	ME-TI (24%)
	SO-MI	70%	SO-RE (11%)
Minor Seventh	SO-FA	54%	SO-RE (13%)
Major Seventh	DO-TI	41%	DO-TE (50%)

*Primary confusions are defined as wrong responses that occurred at least 10% of the time.

The fourth phase of the analysis sought to determine whether there was a common confusion in the case of the descending fifth, and whether common confusions could be identified for other intervals. Cross tabulations were made to show what intervals were asked above and below each pitch, and what responses the students gave. As before, only those intervals in which students correctly identified the first note on the first try were included in the analysis.

Tables 4 and 5 summarize the findings for ascending and descending intervals, respectively. Column one identifies the interval name of the stimulus. Column two gives the stimulus, and column three contains a percentage that shows how often the students answered it correctly. Column four shows primary confusions, where "primary" refers to a confusion that accounted for at least 10% of the student error on a given stimulus. The percentages in parentheses show how often the primary confusions were answered in response to the stimuli.

All of the confusions reported in tables 4 and 5 fall into one of five types of errors that students make in melodic dictation exercises. First is the "Major-Minor Mode" error. This accounts for the confusion of TI for TE and LA for LE that was identified above in Figure 3.

TABLE 5
Primary confusions with descending intervals.

Interval	Stimulus	% Correct	Primary Confusions*
Minor Seconds	DO-TI	84%	None
	ME-RE	90%	None
	FA-MI	83%	None
	LE-SO	90%	None
Major Seconds	DO-TE	79%	DO-TI (15%)
	RE-DO	87%	None
	MI-RE	80%	None
	FA-ME	83%	None
	SO-FA	79%	None
	LA-SO	90%	None
	TE-LE	94%	None
	TI-LA	91%	None
Minor Thirds	DO-LA	35%	DO-TI (30%), DO-SO (22%)
	RE-TI	62%	RE-DO (24%)
	ME-DO	77%	ME-RE (11%), ME-TI (11%)
	FA-RE	56%	FA-ME (18%), FA-MI (12%)
	SO-MI	65%	SO-FA (15%)
	LE-FA	56%	LE-SO (32%)
Major Thirds	MI-DO	77%	MI-RE (12%)
	SO-ME	60%	SO-RE (11%), SO-FA (11%)
	TI-SO	75%	None
Perfect Fourths	DO-SO	56%	DO-LA (19%)
	SO-RE	47%	SO-MI (28%)
Perfect Fifths	DO-FA	41%	DO-SO (22%)
	SO-DO	51%	SO-ME (14%), SO-RE (11%)
Minor Sixths	DO-MI	52%	DO-SO (17%)
	ME-SO	71%	ME-TI (10%)
	SO-TI	63%	SO-TE (11%)
	LE-DO	62%	LE-ME (17%)
Major Sixths	DO-ME	60%	DO-SO (11%)
	MI-SO	67%	MI-TI (14%)
	LA-DO	68%	LA-RE (10%)
Minor Sevenths	DO-RE	48%	DO-TE (12%)
	ME-FA	42%	ME-SO (21%), ME-RE (12%)
	SO-LA	50%	SO-LE (18%), SO-FA (11%)
	LA-TI	77%	None
Major Sevenths	MI-FA	56%	MI-MI (11%), MI-SE (11%)
	SO-LE	82%	None

*Primary confusions are defined as wrong responses that occurred at least 10% of the time.

Second is the "Scalewise Underestimation" error, whereby students tend to underestimate the size of melodic leaps by one diatonic scale degree. The descending perfect fifth falls into this category, as it is primarily confused with the perfect fourth.

Third is the "Inner Chord Tone" error, which is most pronounced in the case of the descending sixth. By studying Table 5, it can be seen that descending sixths are confused with fourths instead of fifths. Since the descending fourth forms a first-inversion triad when added beneath the top note of a descending sixth, one can hypothesize that students hear the implied harmony conjured up by the sixth and answer with the inner chord tone instead of the sixth. This phenomenon was observed for the Major sixth down from DO, with which the insertion of a descending fourth forms a minor tonic triad; the major sixth down from MI, to form a iii chord; the minor sixth down from DO, to form a major tonic triad; the minor sixth down from ME, to form a III+ chord; and the minor sixth down from LE, to form a VI chord. It was not observed for the ascending sixth, where the primary confusion was the perfect fifth in accordance with the scalewise underestimation rule. Inner chord tone errors also account for the answering of SO-ME for the descending fifth SO-DO to form a minor tonic triad in the minor mode (14%), SO-RE for the ascending minor seventh SO-FA from an implied dominant seventh chord (13%), and DO-MI for the ascending fifth DO-SO to form a major tonic triad.

Fourth is a curious "Reversed Inversion" error that affects descending sevenths. Instead of following the underestimation rule and confusing descending minor sevenths with sixths, students confused descending minor sevenths with descending seconds. The students probably heard the inversion of the seventh, but since the interval obviously went down, they answered with a descending second instead of an ascending second. This occurred as a significant confusion for minor sevenths starting on DO, ME, and SO. It was also the most frequent confusion for minor sevenths down from LA, but students answered LA down to SO for LA down to TI only 6% of the time. The significantly higher score (77%) that students achieved on LA-TI as compared to the other descending sevenths, which were scored from 42% to 50%, shows how scalar position can have a profound impact upon the perception of an interval. The descending seventh LA-TI was always followed by DO, and the strong tonic pull of the ascending second TI-DO accounts for the higher scores of the LA-TI descending seventh.

Fifth, students made "Scalewise Overestimation" errors in confusing descending thirds with fourths and descending major sevenths with octaves. This error pattern did not affect the perception of ascending intervals, and it always occurred in conjunction with underestimation errors; that is, intervals affected by scalewise overestimation were also affected by underestimation.

The final phase of this study consisted of an analysis of student learning styles. Table 6 shows for each unit in the curriculum the number of exercises completed, the average speed of dictation, the average response time per exercise, the average number of plays per exercise, and the percentage of notes answered correctly on the first try. The unit in which students spent more time than any other was number 5, "Melodies with Leaps from I and V in Major Keys" (17.4%). This unit had the lowest percentage of notes correct during training (60%). Also difficult was the last unit, "More Angular Melodies in Minor Keys," in which the students spent 12.7% of their time and worked at an average score of 61%. None of the other units accounted for more than 6% of the exercises.

It was expected that as students progressed through the curriculum, they would gradually lower the speed of dictation, as they had in the rhythm (Hofstetter, 1981, p. 274) experiment. However, the mean speed of dictation increased slightly from 39 to 43 beats per minute. On the average, students asked for each melody to be played twice. This was lower than had been expected, as was the average response time of 31 seconds per exercise.

Table 7 shows the results of a multiple correlation that measured the interactions of the speed of dictation, the mode (major or minor) of the exercises, the average number of accidentals per exercise, the average number of plays per exercise, response time, pitch boxes, solfeggio syllables, and percent correct during training. Neither speed, mode, nor key signature correlated highly with percent correct during training. Nor did the use of pitch versus solfeggio answer boxes, although

TABLE 6
Analysis of student learning time.

Unit	Title	Number of Exercises	% of Total Exercises	Average Speed		Average Response Time		Plays per Exercise		Student Performance	
				Beats per Minute	SD	Seconds per Exercise	SD	Average Number	SD	% Correct	SD
1	Short Scalewise Melodies in Major Keys	171	4.1	39	9.0	22	14.0	1.3	0.7	71	19.1
2	Longer Scalewise Melodies in Major Keys	124	3.0	39	9.6	30	13.8	1.9	1.1	77	12.0
3	Longer Scalewise Melodies: Solfeggios	171	4.1	39	8.9	25	13.1	1.7	1.0	68	19.5
4	Leaps from the Tonic Triad in Major Keys	180	4.3	39	11.2	29	15.4	1.8	1.3	64	19.3
5	Melodies with Leaps from I and V in Major Keys	729	17.4	41	8.8	33	14.8	2.3	1.3	60	18.9
6	Leaps from I and V: Solfeggios	178	4.3	41	7.6	17	7.2	1.5	0.7	68	15.5
7	Leaps from I, IV, V, and VII in Major Keys	155	3.7	36	11.2	49	22.9	3.2	1.5	65	20.1
8	Short Melodies in Minor Keys	114	2.7	37	11.3	31	16.1	2.1	1.0	68	16.3
9	Longer Melodies in Minor Keys	222	5.3	40	9.3	38	18.0	2.3	1.1	68	15.9
10	Longer Melodies in Minor Keys: Solfeggios	218	5.2	41	7.1	27	14.0	1.9	1.1	66	18.3
11	Leaps of a Sixth in Major Keys	211	5.0	42	6.1	29	14.3	2.0	1.0	66	16.8
12	Leaps of a Sixth in Minor Keys	210	5.0	43	4.7	32	15.9	2.0	0.9	68	14.3
13	Leaps of a Seventh in Major Keys	143	3.4	42	5.0	27	13.7	1.9	1.0	75	14.3
14	Leaps of a Seventh in Minor Keys	205	4.9	42	5.0	26	12.5	1.7	0.8	69	16.0
15	Sixths and Sevenths in Major Keys: Solfeggios	165	3.9	43	4.6	24	11.9	2.1	1.2	70	15.7
16	Sixths and Sevenths in Minor Keys: Solfeggios	246	5.9	43	5.7	26	11.4	2.2	1.1	64	17.2
17	More Angular Melodies in Major Keys	207	5.0	42	5.6	44	23.1	3.2	1.5	64	19.0
18	More Angular Melodies in Minor Keys	531	12.7	43	5.1	34	16.4	2.4	1.2	61	17.8
		4180	100.0	41	7.9	31	16.7	2.2	1.2	66	18.0

TABLE 7
Multiple correlation of melodic dictation variables.

	Average Speed per Exercise	Major or Minor Mode	Accidentals per Exercise	Plays per Exercise	Response Time per Exercise	Pitch Boxes	Solfeggio Boxes	% Correct During Training
Average Speed per Exercise	1.0	0.08	0.03	-0.03	-0.17	-0.02	0.02	-0.07
Major dr Minor Mode		1.0	0.12	0.0	0.01	-0.05	0.06	-0.02
Average Number of Accidentals per Exercise			1.0	0.02	0.07	-0.33	0.33	-0.08
Average Number of Plays per Exercise				1.0	0.71	0.13	-0.12	-0.18
Average Response Time per Exercise					1.0	0.22	-0.21	-0.39
Pitch Answer Boxes						1.0	-0.98	-0.08
Solfeggio Answer Boxes							1.0	0.05
Percent Correct During Training								1.0

students took longer to answer with pitch boxes. The fact that it requires two touches to enter a pitch with pitch boxes but only one touch with solfeggio boxes accounts for this difference in response times. Even though the number of accidentals per exercise was negatively correlated (-.33) with pitch boxes and positively correlated (.33) with solfeggio answer boxes, it still took less time for students to answer with solfeggio answer boxes. This reinforces the notion that having to think about letter names adds a notational layer to the more immediate solfeggio response.

As one would expect, response time was highly correlated (.71) with the number of plays per exercise; the more often students asked to hear an exercise, the longer they spent trying to answer it. Listening again was somewhat negatively correlated (-.18) with success, as was response time (-.39); when students got an exercise right, they tended to answer it promptly on the first attempt.

DISCUSSION

Four anomalies merit further discussion. First, why do students increase instead of decrease the speed of dictation as melodies become longer and more complex? In the GUIDO rhythm study (Hofstetter, 1981) students gradually lowered the speed of dictation as the rhythms increased in length and complexity; the expectation was that similar behavior would be observed for melodies. However, the students chose lower initial dictation speeds for melodies than for rhythms. The mean speed for Unit 1 of melodies was 39 beats per minute, whereas it was 115 beats in Unit 1 of rhythms. The reason why students increased the speed of melodic dictation probably has to do with the fact that if the dictation speed is set too slow for a long melody, students tend to forget the beginning notes before the final notes are played. Increasing the dictation speed allows the final notes to be heard within the perceptual time frame required to remember both the beginning and the ending of a long melody.

Second, why do repetitions of the stimulus not lead to increased achievement scores? The mean number of repetitions in this study was one per melodic dictation exercise. Students listened to each melody twice on the average, and the standard deviation was one additional hearing. Listening again had little effect on achievement. In fact, the average number of plays per exercise was somewhat negatively correlated (-0.18) with student achievement. After the students listened to an exercise three times, they either knew it or not. Perhaps the reason why further repetitions did not help is because of the ingrained melodic confusion tendencies identified in Tables 4 and 5.

Third, why do students perform poorly on the descending fifth, while they excel on the ascending fifth? The same confusion patterns are found for both ascending and descending fifths, where students either underestimate the size of the interval by a diatonic scale degree or fill it in with an implied chord tone. For some unknown reason, the descending interval is more difficult to hear than its ascending counterpart.

Finally, why do students hear implied chord tones for descending sixths but not for ascending sixths? Evidently, the descending sixth activates a harmonic perception mechanism that is not triggered by the ascending sixth. Students mentally arpeggiate the implied harmony when they hear a descending sixth, whereas the ascending sixth does not conjure up such harmonic implications.

REFERENCES

Hofstetter, Fred T. "Computer-Based Recognition of Perceptual Patterns in Harmonic Dictation Exercises," Journal of Research in Music Education, 26 (1978), 111-119.

Hofstetter, Fred T. "Evaluation of a Competency-Based Approach to Teaching Aural Interval Identification," Journal of Research in Music Education, 27 (1979), 201-213.

Hofstetter, Fred T. "Computer-Based Recognition of Perceptual Patterns in Chord Quality Dictation Exercises," Journal of Research in Music Education, 28 (1980), 83-91.

Hofstetter, Fred T. "Computer-Based Recognition of Perceptual Patterns and Learning Styles in Rhythmic Dictation Exercises," Journal of Research in Music Education, 29 (1981), pp. 265-277.

Hofstetter, Fred T. "Automatic Syntactic Question Generation for Computer-Based Music Learning Systems." Submitted for publication in the Journal of Computer-Based Instruction.

Tukey, John W. Exploratory Data Analysis (Reading: Addison-Wesley Publishing Company, 1977).

ADCIS Proceedings, November 1986

PILOT as a User Interface to a Personal File System[1]

John A. Starkweather

University of California, San Francisco

Although PILOT is most commonly used as a way to program interaction for instruction or testing, its ability to initiate other programs makes it useful as a control mechanism, analogous to batch files in MSDOS or PCDOS, submit files in CP/M, or shell scripts in UNIX. PILOT proved to be an effective method for development of interaction with users of a bibliographic filing and retrieval system developed for use of medical research investigators.

Why was a special file system developed?

The overall intent of the project was to see if bibliographic information files maintained by a research group would also satisfy the bibliographic file needs of individual users within the group. We proposed to support both a group file for co-investigators working in the same subject area and a personal file for each of the individuals involved. We then planned to study how the individual's file might proceed to diverge from the common group file. Different members of a research team may have special interests that diverge in different amounts from the common interests of the group.

These project goals required development of a system that could store information into either the user's individual file or into both the individual file and at the same time into a group file. The system also needed to capture information about user activity, whether in entering or in retrieving information, in appropriately labeled logs. It was desireable to protect users from the need for detailed knowledge about these mechanisms by the use of menus and screen displays that would prompt for information to be filled in.

The system without PILOT

The filing system itself was not unusual. Entries consisted of lines of text with accompanying codes to indicate reference information, such as author, title, source, and comments or keywords. A normal text editor could be used to create entries in the proper format and with the required codes, so that it was possible for the system to be built and tested without the involvement of PILOT. A sample item is shown in Figure 1. Within a UNIX timesharing system an entry could be stored in a file accessible to a group as well as in an individual's personal file. But there were two major problems: there was no easy way to guard against input errors, and users needed to learn quite a lot about UNIX commands in order to accomplish the storage of information and make requests for retrieval and output formatting.

Entry of information using an editor (one not especially designed for the purpose) allowed errors to occur that violated the format and coding requirements. It was necessary to create a validation program that users could run to find such errors and allow them to be corrected. If entry of data elements could be prompted and guided, then formatting and the addition of codes could be added automatically.

For a knowledgeable user, existing UNIX programs could be invoked in sequence to convert entries to single long lines, search for text patterns, and convert back to printable output. In the following example, a file named "dan.pers" would be searched for the pattern "stress response" and those items in which the pattern was found would be stored in the file "dan.dest".

```
contract < dan.pers | grep -i "stress response"|
 expand > dan.dest
```

A program "contract" accepts input from dan.pers, converts the multiple lines of a stored item into a single long line, sends its output to "grep" which is a UNIX utility program that looks for the occurrence of the text string "stress response". The lines in which the text string is found (the output of grep) are sent to another program "expand" which converts them back to multiple line items and sends the results to dan.dest.

---

[1] The work reported here was supported by U.S. Public Health Service grant R01 LM 03800, an extramural grant of the National Library of Medicine.

To operate at this level, our users would need to learn the UNIX editor, verify the accuracy of their entry, and then be able to prepare commands to the UNIX system such as the one in this example.

PILOT can save the user from errors and complexity.

We designed an entry system that would not require the user to learn more about UNIX than what was required to sign on to the computer, but one that would appropriately prompt for the entry of bibliographic citations and notes. The system is written in PILOT, because of the ease with which human-machine interaction can be developed. The PILOT program ran within the UNIX operating system environment, making use of existing UNIX tools whenever possible for such operations as file management, search of text patterns, sorting, and the collection of logs of user activity. User of PILOT and the UNIX operating system also allowed us to adapt the program to run in a functionally identical manner on three different computers: a DEC PDP-11/70, a VAX 750, and a much smaller PDP-11/24. The menu-driven input was tailored for journal articles, books (or chapters of books), and informal communications (notes or memoranda). The programming went through three cycles of investigator comments and revisions before we considered it ready for use by our collegues who were acting as subjects. Part of this development was carried out by writing a functionally similar program in PILOT on a microcomputer. It presented the same menu screens and accepted input for storage onto floppy disks which could later be transmitted to UNIX. This provided some cost saving in the development of group files in comparison to online use of Computer Center equipment.

We emphasize the need for a simple, easy-to-use interface between the system and its users. PILOT provided a rapid way to develop such an interface. We did not believe that infrequent users could be expected to learn complex procedures to use bibliographic files. The use of menus with fields to be filled in provides one way to prompt users in an obvious way so that very little instruction is required. In our system, a main menu gives the user the choice of seven options: to add an article in a journal, to add a book or a chapter in a book, to add a personal note or other unclassified item, to retrieve information, to erase items from a personal file, to revise items from a personal file, or to communicate with other users with the UNIX mail system. Figure 2 is an image of the main menu, and Figure 3 shows the PILOT code for this display. The mail option was added when it became obvious how important the exchange of information was to cooperative research efforts. The selection of one of these options from the main menu causes a new menu to be displayed that is tailored to the specific chosen task. Figure 4 shows the display used for entering information about a journal article, and Figure 5 is the equivalent display for books or a book chapter. Each time that a user adds an item, a further option is presented, that of whether the item should be added to the personal file only, or to both the personal and group files. (We required that any addition to the group file must also be in that user's personal file.) The group file was expected to grow with additions from other individuals in the group, and we expected to be able to measure "divergence" as the group file became a collection representing items chosen for sharing by group members and as the various personal files represented their individual interests.

When a user wishes to retrieve information from either the group or personal file, the "lookfor" menu is chosen. This menu, shown in Figure 6, allows the choice of what file will be retrieved from, either one's personal file (presented as a default), the group file, or a file resulting from a prior search. One may then look for a keyword, a phrase, or in fact any sequence or pattern of characters that might be present in the stored item. We used a mechanism that will retrieve any character string without regard to whether it occurs in the author name, journal name, title, in added notes, or in any other field. We find that users often use keywords added in the annotation field, which allows free-form notes, as the basis for retrieval. The citation or citations found are shown on the screen and automatically stored in a separate file for further use if desired.

The menus for erasing or revising personal file entries are arranged in similar manner to the menu for retrieval. They allow the same choice of files for source and destination, and the functions of erasing or revising are of obvious utility in the maintenance of one's references. By using the same internal format as that developed by another group on our campus, we are able to format a file of citations into the formats required by a number of leading biomedical journals. The system allows one to specify such formats with a script which is not difficult to modify for new requirements that may arise. For example, Figure 7 shows two items formatted for two different journals.

Conclusion

PILOT provided a comfortable data entry method that was readily understandable with minimal instruction. By prompting entry in a consistent fashion, it was no longer necessary to use a verification program to avoid format errors. Complex procedures were accomplished by other programs but users did not need to be concerned about their complexity. PILOT was easily modified to meet user needs as the system was developed.

```
Horowitz MJ
(1973) American Journal of Psychotherapy 27:506-515
*Phase-oriented treatment of stress response syndromes
+1131 short term psychotherapy

Attkisson CC, Hargreaves WA, Horowitz MJ, Sorenson J
(1978)B
*chap: Evaluation: Current Strengths and Future Directions|
book: Evaluation of Human Service Programs|
eds: C. C. Attkisson, et al|
place: New York|
pub: Academic Press|
pp: 465-478
+1155 review
```

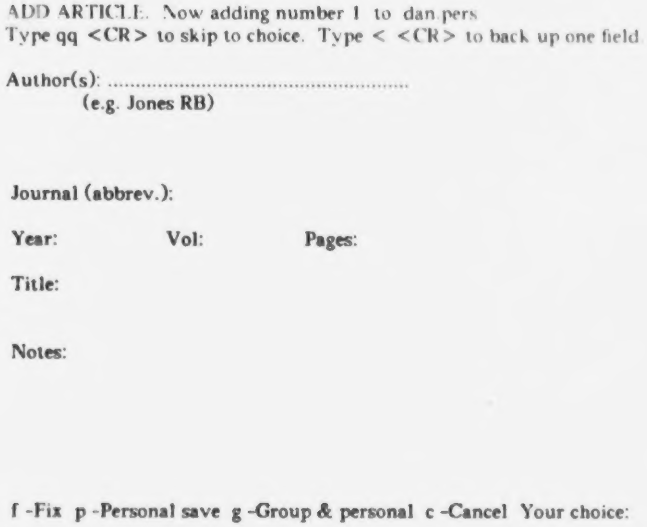
Figure 1

```
                    MAIN MENU

a - add Article in a journal

b - add Book or chapter in a book

p - add Personal note or other unclassified item

l - Lookfor menu

e - Erase menu

r - Revise menu

m - Mail

c - Change to not require <CR> for choice selection

q - Quit

Your choice (pick a,b,p,l,e,r,m,c or q): _   (<CR> needed)
```

Figure 2

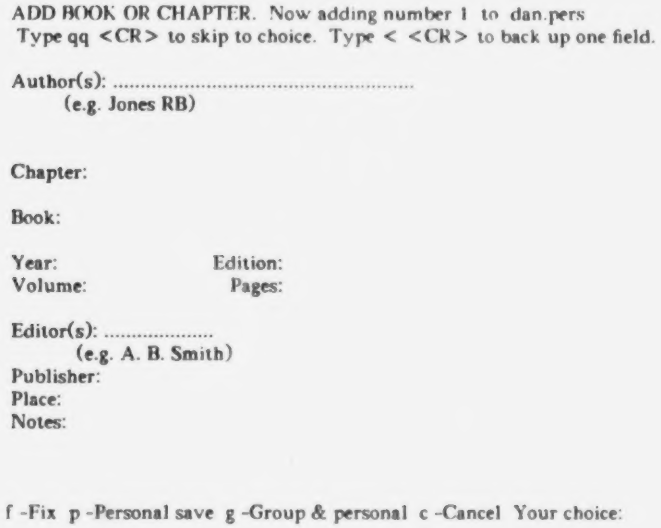
```
t:                         MAIN MENU
t:
t:  a - add Article in a journal
t:
t:  b - add Book or chapter in a book
t:
t:  p - add Personal note or other unclassified item
t:
t:  l - Lookfor menu
t:
t:  e - Erase menu
t:
t:  r - Revise menu
t:
t:  m - Mail
t:
t(crflag = 1):   c - Change to not require <CR> for choice selection
t(crflag = 0):   c - Change to require <CR> for choice selection
t:
t:  q - Quit
t:
t:Your choice (pick a,b,p,l,e,r,m,c, or q): _ ( <CR> needed)
```

Figure 3



```
ADD ARTICLE.  Now adding number 1  to  dan.pers
Type qq <CR> to skip to choice.  Type < <CR> to back up one field.

Author(s): ....................................................
          (e.g. Jones RB)


Journal (abbrev.):

Year:              Vol:              Pages:

Title:


Notes:




f -Fix  p -Personal save  g -Group & personal  c -Cancel  Your choice:
```

Figure 4



```
ADD BOOK OR CHAPTER.  Now adding number 1  to  dan.pers
 Type qq <CR> to skip to choice.  Type < <CR> to back up one field.

Author(s): ....................................................
        (e.g. Jones RB)


Chapter:

Book:

Year:                    Edition:
Volume:                    Pages:

Editor(s): ....................
          (e.g. A. B. Smith)
Publisher:
Place:
Notes:


f -Fix  p -Personal save  g -Group & personal  c -Cancel  Your choice:
```

Figure 5

```
                    LOOKFOR MENU

v - View the entries in dan.pers
c - Count the entries in dan.pers

a - Author sort of dan.pers into dan.pers.author
y - Year sort of dan.pers into dan.pers.year
j - Journal sort of dan.pers into dan.pers.journal


l - Look for:          (keyword or phrase)

s - change Source file to:              (from: dan.pers)
                   to:  filename   to see the group bibliography

d - change Destination file to:           (from: dan.dest)

u - Unix command:


Your choice:             (m - to MAIN MENU)
                         (e - to ERASE MENU)
                         (r - to REVISE MENU)
```

Figure 6

REFERENCES

Attkisson, C. C., Hargreaves, W. A., Horowitz, M. J., & Sorenson, J.(1978). Evaluation: Current Strengths and Future Directions. In C. C. Attkisson, et al (Ed.) *Evaluation of Human Service Programs*. New York: Academic Press, 1978.
Horowitz, M. J. (1973). Phase-oriented treatment of stress response syndromes. *American Journal of Psychotherapy*, 1973, *27*, 506-515.

1. M.J. Horowitz, *American Journal of Psychotherapy*, *27*, 506 (1973).
2. C.C. Attkisson, W.A. Hargreaves, M.J. Horowitz and J. Sorenson, in *Evaluation of Human Service Programs*, C. C. Attkisson, et al, Ed. (New York, Academic Press, 1978), pp. 465-478.

Figure 7

The Evaluation of an Automated Test Delivery System (CMI) for the United States Army Engineer School

Captain Tracie Carter, United States Army Engineer School

ABSTRACT

In the fall of 1985, the United States Army Engineer School conducted a study of school requirements for an on-line testing program. The school then evaluated four computer based testing packages against those requirements. The four systems tested were the Army's Automated Information Management System, the PLATO system using the TUTOR language, a test driver developed by Sheppard Air Force Base and the Generic Test Delivery System developed by Control Data Corporation. The results of this evaluation determined whether the Engineer School would adopt computerized testing for its officer courses. Generic Test Delivery System, TUTOR, and PLATO are all copyrights of Control Data Corporation.

INTRODUCTION

During the period August 1, 1985 through October 21, 1985 a study of the United States Army Engineer School requirements for an automated test delivery system and the methods available to meet those requirements was conducted by Technology Training Division, Department of Military Engineering. The evaluation was conducted using four potential test delivery systems: the Automated Information Management System (AIMS), the PLATO authoring system (TUTOR), a test driver developed by Sheppard Air Force Base (SAFB), and a Control Data Corporation product entitled Generic Test Delivery System (GTDS).

The United States Army's Engineer School provides training for over 3000 Army officers and noncommissioned officers per year. The training consists of various technical engineering subjects and many basic army skills such as English, staff organization, and tactics. Much of this training is supported by the use of Computer Based Training as the selected medium for teaching. The majority of the Computer Based Training is based on Control Data Corporation's (CDC) PLATO system. This is supported by a 60 terminal computer laboratory using CDC Viking terminals with a CDC Cyber 830 supermini computer.

The purposes of the evaluation were to:

a. Identify the requirements of the Engineer School for an automated test delivery system.

b. Analyze the capabilities and limitations of the four systems evaluated.

c. Determine the best system for use at the U.S. Army Engineer School.

The evaluation was limited to only those test delivery systems that were readily available at little or no cost to the government. GTDS was provided by Control Data Corporation for evaluation purposes.

STATEMENT OF NEED

Due to the heavy instructional management requirements in the U.S. Army Engineer School, the automation of testing is critical in order to minimize the resources used in support of test item generation, test administration, test scoring, test item analysis and instructional sensitivity decisions based upon evaluation data. The U.S. Army Engineer School currently employs paper-based examinations in the assessment of student performance throughout the total curriculum. The current test delivery system involves high printing costs and excessive time demands for preparation, monitoring, test analysis, and student remediation. Additional resources are required to provide adequate test security.

It was believed that the use of an on-line testing mechanism would reduce the printing costs, and the instructor time required for preparation, test analysis and student remediation and would create an enhanced test security methodology.

ENGINEER SCHOOL REQUIREMENTS

The following capabilities of a test delivery system were deemed to be essential to the Engineer School.

STUDENT REQUIREMENTS:

a. The student must receive immediate feedback as to pass/fail and score.

b. The student must be able to review missed questions as a learning tool.

c. The test driver must be user-friendly requiring no prior computer training of the student.

INSTRUCTOR AND MANAGEMENT REQUIREMENTS:

a. The test driver should be easy for instructors to use, i.e., the instructor must be able to input his own test.

b. The instructor must be able to determine student performance data without interpretation. The test driver must provide meaningful data for management decisions, to include test item analysis.

c. The instructor should be able to use the test as a training mechanism, i.e., provide feedback to students and improve instruction.

d. The test driver must save time and effort for the instructor and manager.

e. The test driver must be secure. First by being accessible only by those with a need to know and second only accessible to students when taking an examination, with no reentry allowed.

SYSTEMS DEFINITIONS

The following is a brief description of the four test delivery systems evaluated:

a. Automated Information Management System (AIMS). The AIMS system is an Army owned operating system which runs on the VAX 11/750. It is used at the Engineer School to maintain student records. The AIMS test system was developed to give an instructor the opportunity to pool a number of test questions to be placed on an exam and then print it out. The system is The results of this comparison are shown at Figure 1. designed to deliver paper based exams and then score them with a mark sense scanner. There are currently no graphic capabilities available. This is the only system examined that does not deliver the test on-line to the student.

b. The PLATO authoring system (TUTOR). The TUTOR language is a multipurpose language designed for instructional purposes. TUTOR can be used for test delivery purposes by an experienced programmer.

c. The Sheppard Air Force Base test driver (SAFB). This test delivery system was designed by a programmer at Sheppard Air Force Base to deliver up to 50 questions on a multiple choice test. It is based on the TUTOR programming language but does not require the user to develop the general utilities for a test. In order to use SAFB the instructor must have a working knowledge of the TUTOR programming language.

d. The Generic Test Delivery System (GTDS). GTDS is a commercial product provided to the Engineer School by Control Data Corporation on an evaluation basis for 60 days. The GTDS was designed to be a programmerless testing system with the ability to provide up to 250 multiple choice questions to the student. GTDS is based on the TUTOR language and has all utilities built in. The instructor is not required to have prior computer training before using GTDS to prepare an exam.

The studied systems were evaluated by testing the capabilities of each system against the requirements of the Engineer School. The AIMS was evaluated by talking to personnel that were familiar with the testing capabilities of the system and had experience using it. The AIMS system was eliminated early because it did not meet any of the student requirements nor did it meet a substantial number of the instructor and management requirements.

TUTOR and SAFB were evaluated by Technology Training Division personnel using previously developed exams and by inputting new exams to test the amount of time required to input the exams. GTDS was evaluated by developing two exams for use by students in three classes of the Engineer Officer Advanced Course. Two other exams were developed using GTDS in order to ascertain the ability of non-programmers to use the system.

A series of questions was developed for the purpose of comparing the four systems. The questions were divided into three categories: student issues, instructor issues, and analysis issues.

RESULTS

a. AIMS: The AIMS did not address student issues because it was not developed as an on-line testing system. For this reason, AIMS does not save the instructor time in test preparation and grading. AIMS also did not save exam printing costs and security was difficult. All systems required student grades to be entered manually into the AIMS gradebook.

b. TUTOR: TUTOR language could be used to satisfy every known requirement; however, the length of time required to learn the language makes its use by other than experienced programmers impractical.

c. SAFB: The SAFB system fulfilled the student requirements and instructor analysis requirements; however several limitations were found when addressing instructor issues. The SAFB has an upper limit of 50 questions per exam. A survey of the number of questions per exam for the Engineer Officer Advanced Course shows that exams had as many as 115 questions. Additionally, there is only one form of the test on-line, there is no capability to create alternate forms of the same exam. Therefore, all students receive the same exam and test security could eventually be jeopardized.

Another drawback of the SAFB is the requirement for the instructor to have a working knowledge of the TUTOR language. This is particularly important when using the graphics capabilities of PLATO. The major advantage of the Sheppard Air Force Base test driver is the ability of the test driver provide the instructor with meaningful statistical data and provide the student with the ability to review the test for missed questions.

d. GTDS: The GTDS fulfilled all student and instructor requirements except the option to allow students to review missed questions as a learning tool. In the area of analysis, GTDS provides some statistical data but does not allow for separating failures, and compiling some course statistics. In addition, GTDS is limited in the amount of information it can provide the instructor for test management purposes. The advantages of GTDS include the programmerless qualities of the system and the ability to test up to 250 questions per exam with several forms testable simultaneously. This allows the instructor to maintain total control of the security and management of the examination.

BENEFITS AND SAVINGS

There are many benefits to be achieved by the use of on-line testing. These benefits fall into several categories including benefits to students, benefits to instructors, and security benefits.

a. Student Benefits: On-line testing provides students with the ability to receive immediate feedback of their score and the questions that were missed. It can also provide the correct response for missed questions. On test drivers that allow for a follow on lesson, students can be taken directly to a remedial lesson. These aspects of on-line testing allow exams to be used as remediation tools.

b. Instructor Benefits: On-line testing saves instructor time in grading exams. It is estimated that it takes an instructor approximately 90 minutes to grade 60 exams containing 50 multiple choice questions. Given the number of multiple choice exams administered at the Engineer School per year there is a significant amount of effort to be saved by using on-line testing systems. An estimate of the amount of time spent grading exams annually is in Figure 2.

Other benefits to instructors can be derived from analysis of the statistical data compiled by the test driver. On those systems having test item analysis, an instructor can determine the relative merit of each test question and also judge the effectiveness of his teaching. Because of the time and effort required to generate this information by hand most instructors use only a very limited form of test item analysis for paper based tests. An estimate of the cost to perform the test item analysis is in Figure 3. The ability to grade exams and perform test item analysis for an instructor make on-line testing systems highly beneficial.

c. Security Benefits: Securing paper based exams is a major burden because of the volume of paper and the number of exams given at the Engineer School. An on-line testing system eliminates the need for one copy of the exam for each student per class. Copies still have to be maintained but not to the same extent. An analysis of the potential savings in printing costs is at Figure 4.

Other security benefits are derived from the fact that in order to access test information a user must have the proper security codewords and have access to the proper authoring group. Students do not have access to either.

CONCLUSIONS AND RECOMMENDATIONS

In Figure 5 there is an evaluation of the four systems and whether they met the requirements of the Engineer School. This comparison shows that GTDS and SAFB are the most acceptable, however each has flaws that must be overcome. For GTDS the drawbacks were its failure to provide adequate student feedback and limited test item analysis. For SAFB the drawbacks were the need to have a working knowledge of TUTOR, the test size limitations, and the ability to create only one form of the exam.

Based on this comparison it was found that most features the Engineer School required were available in two testing systems and the decision was made to adopt on-line testing for Engineer School students. In choosing to use on-line testing the Engineer School is purchasing the Generic Test Delivery System program from Control Data Corporation and has also placed the Sheppard Air Force Base test driver on-line. Since the inception of the program both testing systems have been used extensively. Many of the first tests that were used are now undergoing revisions to reflect some to the enhancements available with on-line testing. In making these revisions, we have discovered an additional benefit to on-line testing. Revisions which were previously made by retyping the entire test and having it reprinted now take a matter of only a few moments to correct on-line. On-line testing has proven to be quite successful at the Engineer School both from the student's and from the instructor's viewpoint. Students appreciate the immediate pass/fail feedback and instructors are enjoying the benefits of automatic grading and test item analysis.

FIGURE 1

SYSTEMS COMPARISON

STUDENT ISSUES	GTDS	SAFB	TUTOR	AIMS
CAN THE STUDENT CHANGE ANSWERS?	YES	YES	ALL ITEMS CAN BE ACCOMPLISHED BY TUTOR WITH VARYING LEVELS OF DIFFICULTY. GTDS AND SAFB ARE BASED ON TUTOR CODE.	AIMS IS A TEST GENERATION AND STORAGE MECHANISM. STUDENTS RECEIVE AIMS EXAMS OFF-LINE AND ARE SCORED BY A MARK SENSE SCANNER.
CAN THE TEST BE TIMED?	YES	YES		
IS THERE A TIME REMAINING ALARM?	YES	NO		
IS THERE A SAFEGUARD AGAINST ACCIDENTAL QUITS?	YES	YES		
DOES THE INDEX SHOW UNANSWERED QUESTIONS?	YES	YES		
CAN THE STUDENT EXIT WITH BLANK QUESTIONS?	YES	NO		
CAN THE STUDENT MARK QUESTIONS FOR REVIEW?	YES	NO		
IS THERE A QUESTION INDEX?	YES	NO		
CAN THE STUDENT REVIEW BEFORE EXITING?	YES	YES	YES	
CAN THE STUDENT MAKE COMMENTS ABOUT THE EXAM?	YES	YES	YES	
DOES THE STUDENT RECEIVE FEEDBACK FOR MISSED QUESTIONS?	NO	YES	YES	NO
IS THERE A SAMPLE QUESTION?	YES	NO	YES	NO
IS THERE AN INTRODUCTION PAGE?	YES	YES	YES	NO
DOES THE STUDENT RECEIVE FEEDBACK WITH HIS SCORE?	YES	YES	YES	NO
WHAT DOES THE FEEDBACK CONTAIN?	SCORE MESSAGE	SCORE PERCENT	SCORE/PERCENT MESSAGE	NONE
DOES THE STUDENT RECEIVE A PASS/FAIL MESSAGE?	YES	YES	YES	NO
DOES THE TEST PROVIDE A FOLLOW-ON LESSON FOR REMEDIATION?	YES	NO	YES	NO
INSTRUCTOR ISSUES	**GTDS**	**SAFB**	**TUTOR**	**AIMS**
HOW MANY POSSIBLE RESPONSES ARE THERE?	7	6	10+	6
IS THERE A QUESTION POOL?	YES	NO	YES	YES
HOW MANY POSSIBLE QUESTIONS ARE THERE?	1000	50	UNLIMITED	1000
CAN THE TEST BE RANDOMIZED?	YES	NO	YES	YES
CAN THE TEST CONTAIN SUBTESTS?	YES	NO	YES	NO
CAN THE TEST CONTAIN SEVERAL CATEGORIES?	YES	NO	YES	NO
ARE THERE A VARIETY OF TESTING FORMATS?	YES	YES	YES	YES
CAN THE QUESTIONS CONTAIN GRAPHICS?	YES	YES	YES	NO
IS THE TEST DRIVER PROGRAMMERLESS?	YES	NO	NO	NO
CAN THE TEST CONTAIN REFERENCE PAGES?	YES	YES	YES	YES
CAN THE INSTRUCTOR SET PASS/FAIL CRITERIA?	YES	YES	YES	YES
IS THE SYSTEM SECURE?	YES	YES	YES	YES
HOW LONG DOES IT TAKE TO TRAIN THE INSTRUCTOR TO CREATE A TEST?	1/2 DAY	1-2 DAYS	6 MONTHS	1/2 DAY
HOW LONG DOES IT TAKE TO INPUT A 50 QUESTION TEST WITH GRAPHICS?	1-2 DAYS	3-4 DAYS	2 WEEKS	1-2 DAYS

FIGURE 1 (Cont.)

ANALYSIS ISSUES	GTDS	SAFB	TUTOR	AIMS
DOES THE TESTER HAVE TEST ITEM ANALYSIS?	YES	YES	YES	YES
DOES THE SCORING INCLUDE RAW SCORE AND PERCENTAGE?	RAW ONLY	YES	YES	PERCENT ONLY
CAN THE INSTRUCTOR INCLUDE A DISCRIMINATION INDEX?	YES	YES	YES	YES
DOES THE TESTER CALCULATE CLASS AVERAGE AND MEAN?	NO	YES	YES	YES
DOES THE TESTER RANK ORDER THE CLASS?	NO	YES	YES	YES
ARE FAILURES EASILY IDENTIFIED?	YES	YES	YES	YES
ARE FAILURES SEPARATED OUT STATISTICALLY?	YES	YES	YES	YES
ARE LOW MISS AND HIGH MISS QUESTIONS FLAGGED?	YES	YES	YES	NO
CAN THE STATISTICS BE COMPILED BY CLASS?	YES	YES	YES	YES
CAN SEVERAL CLASSES BE COMPILED TOGETHER?	YES	YES	YES	NO
CAN QUESTION RESPONSES BE SUMMARIZED BY CLASS?	YES	YES	YES	YES
DOES THE TESTER SHOW INDIVIDUAL RESULTS?	YES	YES	YES	YES
CAN THE RESULTS/OUTPUT BE FORMATTED TO MEET EVALUATION NEEDS?	YES	YES	YES	YES

FIGURE 2

ANALYSIS OF GRADING EFFORT EXPENDED
ANNUALLY BY THE ENGINEER SCHOOL

EXAMINATION GRADING BY CLASS TYPE	EOAC	EOBC	ANCOC
MULTIPLE CHOICE EXAMS PER CLASS	10	22	51
QUESTIONS PER EXAM	50	50	50
STUDENTS PER CLASS	60	60	60
CLASSES PER YEAR	10	20	10
NUMBER OF EXAMS ADMINISTERED PER YEAR	100	440	510
TOTAL EXAMS ADMINISTERED PER YEAR BY THE ENGINEER SCHOOL		1050	

ASSUMPTIONS: IT TAKES ONE INSTRUCTOR 90 MINUTES TO GRADE A 50 QUESTION MULTIPLE CHOICE EXAM FOR 60 STUDENTS.
AN INSTRUCTOR'S AVERAGE SALARY IS $10.00 PER HOUR.
THEREFORE, IT COSTS APPROXIMATELY $15.00 TO GRADE EACH EXAM

FOLLOWING THIS IT COSTS THE ENGINEER SCHOOL APPROXIMATELY $15,700 PER YEAR TO GRADE EXAMS.

FIGURE 3

ANALYSIS OF INSTRUCTOR COSTS TO PERFORM TEST ITEM ANALYSIS

IN ORDER TO PERFORM BASIC TEST ITEM ANALYSIS FOR A PAPER BASED TEST, INSTRUCTORS WOULD PERFORM THE FOLLOWING STEPS:

1) CREATE AN ARRAY IDENTIFYING THE TEST QUESTIONS, THE POSSIBLE RESPONSES AND THE CORRECT RESPONSES.
2) PLOT THE STUDENT RESPONSES INTO THE ARRAY.
3) IDENTIFY THE LOW-MISS AND HIGH-MISS QUESTIONS.
4) ANALYZE THE POSSIBLE REASONS FOR STUDENT WRONG RESPONSES.

THIS PROCESS CAN BE EXTREMELY TIME CONSUMING. FOR A 50 QUESTION MULTIPLE CHOICE TEST IT TAKES APPROXIMATELY 6 HOURS TO PERFORM EVEN THE MOST BASIC TEST ITEM ANALYSIS.

USING THE 1050 EXAMS DELIVERED BY THE ENGINEER SCHOOL IN A ONE YEAR PERIOD, WITH AN AVERAGE SALARY OF $10.00 PER HOUR, IT WOULD COST APPROXIMATELY $60.00 PER EXAM OR $63,000 PER YEAR TO PERFORM EVEN THE SIMPLEST TEST ANALYSIS.

FIGURE 4

ANALYSIS OF PRINTING COSTS EXPENDED BY THE ENGINEER SCHOOL PER YEAR

BASED ON THE FORT BELVOIR ESTIMATED PRINTING COSTS, THE FOLLOWING ARE THE ESTIMATED ANNUAL COSTS FOR PRINTING ENGINEER SCHOOL EXAMS.

BASED ON 1050 EXAM SHOWN IN FIGURE 2, AT 60 COPIES EACH WITH 5 PAGES PRINTED DOUBLESIDED THERE WILL BE 63,000 COPIES PRINTED OR 315,000 PAGES.

IF 1000 COPIES ARE PRINTED AT A TIME THE COST WILL BE $4300 PER YEAR.

IF AN INSTRUCTOR HAS 60 COPIES PRINTED AT A TIME, THE ANNUAL COST WILL BE $6300.

THEREFORE IF PRINTING COSTS ARE VIRTUALLY ELIMINATED THROUGH THE USE OF ON-LINE TESTING THE SAVINGS COULD BE BETWEEN $4000 AND $6000 PER YEAR.

FIGURE 5

ENGINEER SCHOOL REQUIREMENTS

STUDENT REQUIREMENTS	GTDS	SAFB	TUTOR	AIMS
STUDENT RECEIVES IMMEDIATE FEEDBACK	MEETS	MEETS	MEETS	DOES NOT MEET
STUDENT CAN REVIEW MISSED QUESTIONS	DOES NOT MEET	MEETS	MEETS	DOES NOT MEET
SYSTEM IS USER FRIENDLY	MEETS	MEETS	MEETS	DOES NOT MEET
INSTRUCTOR/MANAGEMENT REQUIREMENTS				
SYSTEM IS EASY TO USE BY INSTRUCTORS	MEETS	DOES NOT MEET	DOES NOT MEET	MEETS
TEST PROVIDES MEANINGFUL DATA FOR MANAGEMENT DECISIONS	MEETS	MEETS	MEETS	MEETS
INSTRUCTOR CAN USE TESTER FOR TRAINING MECHANISM	DOES NOT MEET	MEETS	MEETS	DOES NOT MEET
TESTER SAVES INSTRUCTOR TIME AND EFFORT	MEETS	DOES NOT MEET	DOES NOT MEET	DOES NOT MEET
TESTER IS A SECURE SYSTEM	MEETS	MEETS	MEETS	MEETS

Contact: Tracie Carter
542A Marshall
Fort Belvoir, VA 22060
703-664-3953

ADCIS Proceedings, November 1986

Efficient CAI in Engineering

Helen Kuznetsov, Ph.D.

University of Illinois at Urbana-Champaign

ABSTRACT

Efficient CAI in Engineering is analyzed using the experience gained in developing and implementing courseware in Statics and Road Design and Construction. Described are the kinds of topics and teaching techniques which proved to be most valuable in utilizing fully the unique features of the computer medium in presenting and explaining material taught and in judging topics' mastery achieved by students.

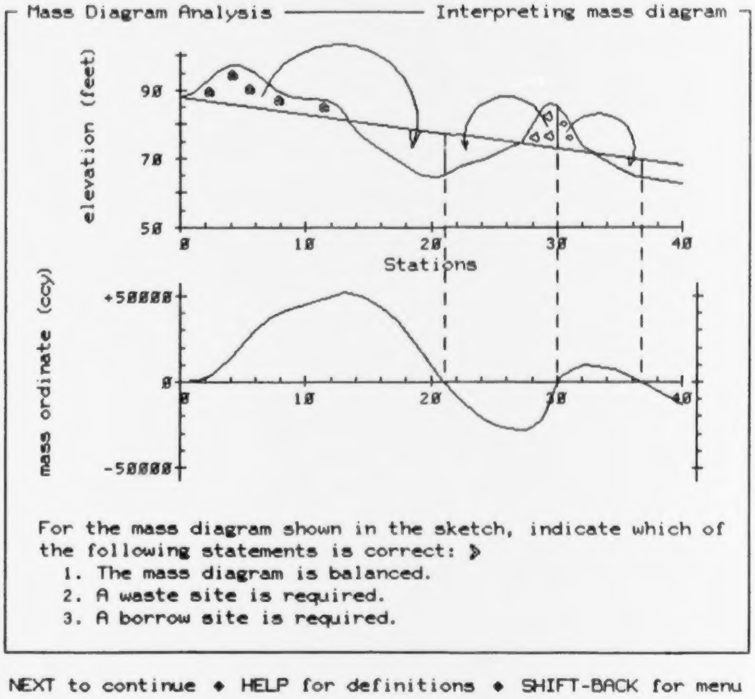
Establishing priorities in topics to be covered by CAI is an important issue considering the amount of investments required for courseware development. Some engineering subjects should be taught in the field on real objects, or in the laboratory by running tests and making real measurements. Unless sophisticated video equipment is used together with CAI, topics of this kind should be avoided in CAI. For example, the description of construction equipment used in road construction could be done better by presenting films made on a real road construction project than by CAI.

The following analysis specifies which features of engineering subjects should be considered in order to produce effective CAI. The idea is to match the requirements of the subject matter with the abilities of computers so that the result is an irreplaceable teaching material which uses the best of the medium and complements other ways of teaching. When comparing different media (textbooks, lectures, or CAI), one should ask which medium is best for teaching certain features of material rather than which medium is generally the best way to teach.

Specifics of Engineering Subjects and Their Affect on CAI

1. Many engineering processes and theories are best explained by their graphical presentation which visualizes the phenomenon and thus helps to comprehend them. For such cases CAI is the best way of teaching the topic. For example, if it is necessary to explain relationships and the practical application of two graphs, only CAI allows the teacher to attract the student's attention to specific features of the graph by flashing the appropriate parts of it. At any given moment the graph in the display may contain only the features important for explanations given in this display. To provide continuity in explanations, the basic graph should not be erased completely when changing the aspect of the presentation. The features of the previous display obstructing the new aspect should be erased, and new details may be added to the sketch.

A good example of this technique is the lesson, "Mass Diagram Analysis," where two graphs, road profile and mass diagram, are presented in the same display simultaneously (Figure 1). While an explanation of how the rising and falling parts of the mass diagram are related to the cut and fill shown



For the mass diagram shown in the sketch, indicate which of the following statements is correct: ⯈

1. The mass diagram is balanced.
2. A waste site is required.
3. A borrow site is required.

NEXT to continue ◆ HELP for definitions ◆ SHIFT-BACK for menu

Figure 1. Interpretation of two related graphs.

in the road profile, an arrow moves along the mass diagram curve, and the words indicating cut or fill flash, when it is appropriate, thus enhancing the learning process.

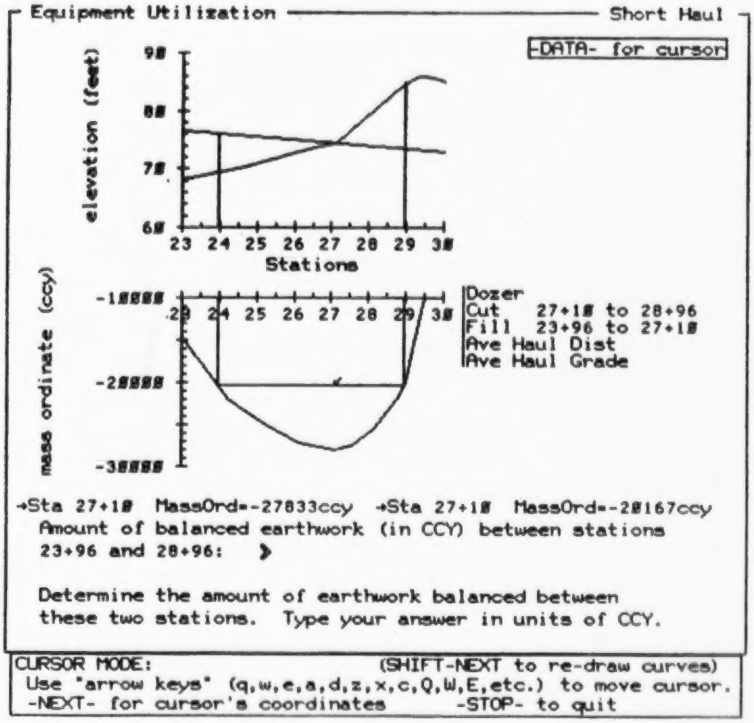
When the concept of a balanced earthwork is explained, an animation is used to move all the stones from the cut area into the fill area. The road is actually built on the screen illustrating how to use the mass diagram for planning the road construction.

When it is necessary to take a reading from the graph, a cursor is introduced, so that a student can move it to the appropriate point on either one of the diagrams and get the reading by pressing NEXT (Figure 2).

To see details of the graph in the vicinity of a specific point of interest, this portion of the graph can be upscaled and presented in the next screen. This approach is used in lesson "Mass Diagram and Equipment Utilization" where it is necessary to find the approximate value of areas under certain portions of the mass diagram and locate the centroids of these areas. When it is necessary to present a longer portion of the graph, it can be downscaled.

All the features of CAI described above show how dynamic the display is in such a lesson. Screen printouts such as Figure 1 or Figure 2, cannot contain everything, and possibly may be misunderstood because of their static nature.

For example, Figure 2 shows the display when the student is using the cursor in order to answer the question at the arrow. The cursor mode explanations below the screen frame appear only after the DATA key is pressed. The



```
Dozer
Cut    27+10 to 28+96
Fill   23+96 to 27+10
Ave Haul Dist
Ave Haul Grade

→Sta 27+10  MassOrd=-27833ccy  →Sta 27+10  MassOrd=-20167ccy
  Amount of balanced earthwork (in CCY) between stations
  23+96 and 28+96:   ❯

  Determine the amount of earthwork balanced between
  these two stations.  Type your answer in units of CCY.

CURSOR MODE:                       (SHIFT-NEXT to re-draw curves)
 Use "arrow keys" (q,w,e,a,d,z,x,c,Q,W,E,etc.) to move cursor.
 -NEXT- for cursor's coordinates        -STOP- to quit
```

Figure 2. Blow-up of the graphs for detailed analysis and a cursor mode in usage.

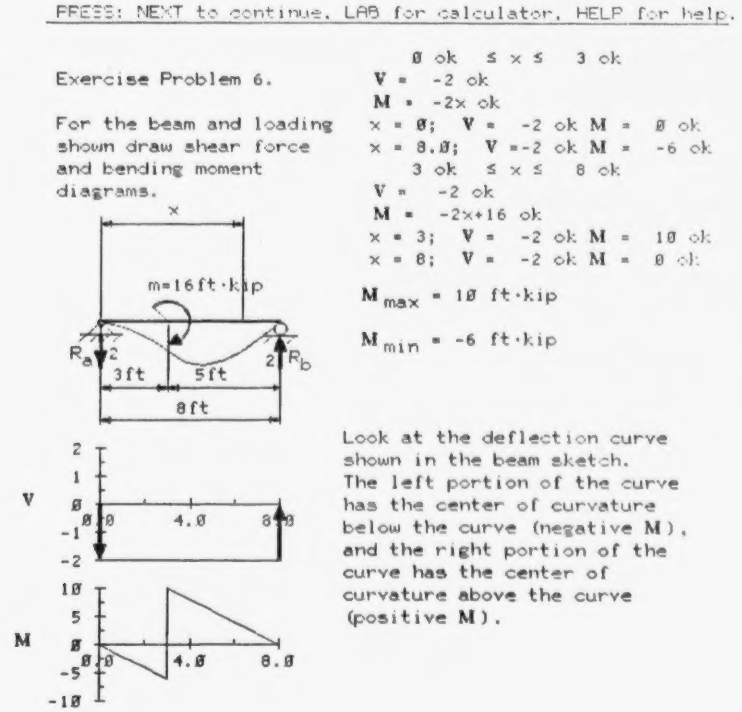
top line below the graphs states stations and mass ordinates of the two points chosen by the student. (The cursor is still located at the second point.) These sophisticated tools and methods of judgment are available only in CAI.

2. When describing any engineering problem by a sketch, a CAI designer should use all the rules and conventions of Engineering Graphics. For example, the sketches have to be scaled to reflect real objects.

To create an accurate picture of the phenomenon presented by the sketch, an algorithm describing the phenomenon can be used for drawing the sketch by computer. This approach is used for drawing a beam and loading sketch, shear force and bending moment diagrams, and the deflection curve of the beam in statics lessons (Figure 3). The data for each problem is a set of values for variables describing the beam geometry and loading conditions (the length, locations of the supports and the load, load characteristics). The sketch is composed by a computer on the basis of the data. The diagrams are the graphs of the appropriate functions drawn by computer after the function entered by a student is judged ok. The deflection curve function is generated by the algorithm which considers both the load and the type and location of the supports.

No other medium but the computer can give an accurate graph of a function, entered by a student, in such a short period of time. CAI is irreplaceable in this situation.

3. Engineering subjects differ from theoretical sciences like mathematics in dealing with reality instead of pure science. Real objects can be described numerically only to



PRESS: NEXT to continue, LAB for calculator, HELP for help.

Exercise Problem 6.

For the beam and loading shown draw shear force and bending moment diagrams.

0 ok ≤ x ≤ 3 ok
V = -2 ok
M = -2x ok
x = 0; **V** = -2 ok **M** = 0 ok
x = 8.0; **V** =-2 ok **M** = -6 ok
3 ok ≤ x ≤ 8 ok
V = -2 ok
M = -2x+16 ok
x = 3; **V** = -2 ok **M** = 10 ok
x = 8; **V** = -2 ok **M** = 0 ok

M_{max} = 10 ft·kip

M_{min} = -6 ft·kip

Look at the deflection curve shown in the beam sketch. The left portion of the curve has the center of curvature below the curve (negative **M**), and the right portion of the curve has the center of curvature above the curve (positive **M**).

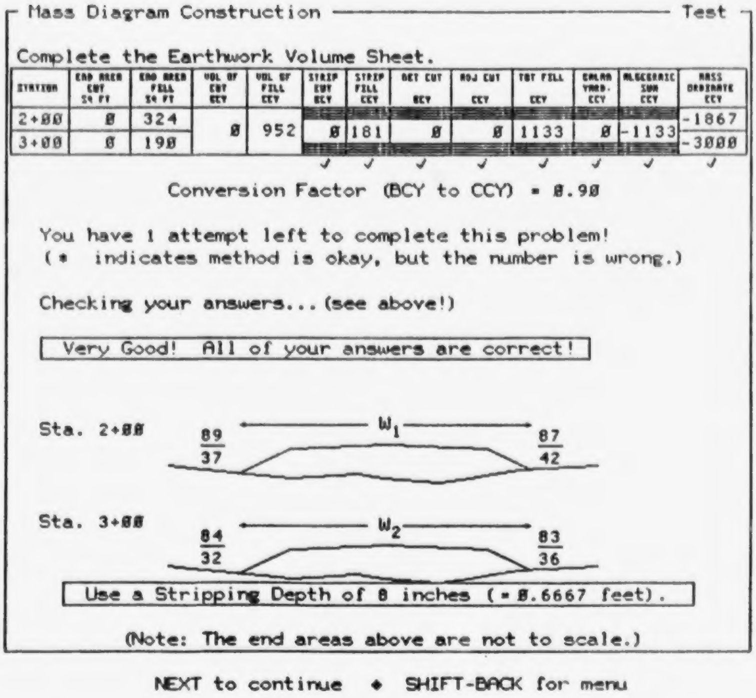
Figure 3. Graphical presentation of a problem and the solution.

some degree of accuracy because there are no two absolutely identical objects in real life. The accuracy of calculations should not exceed the accuracy of measurements providing the information about the objects involved. This means that there should be appropriate tolerances incorporated in the judging of numerical answers in CAI and the computer calculated results should be rounded.

4. In all engineering problems the reasonable values of the variables involved are restricted by the physical nature of the problem. This means that the data for engineering problems should not be generated absolutely randomly. (This tendency was observed when people without an engineering background were involved in the lesson design.)

Several methods for providing different data for the test problems for each student were developed:

a. One approach is to introduce a relatively small disturbing factor applied to the basic set of numerical data which describes a real situation for the problem in question. The disturbing factor itself may be randomly generated within the limits indicated. For example, a disturbing factor between .9 and 1.1 was introduced in lessons dealing with road profile and mass diagram presentations. Multiplying the original real data by a different disturbing factor for each student, produces a new data set for a test problem each time the student enters the test. The profile and diagram are drawn according to the data.



Mass Diagram Construction — Test

Complete the Earthwork Volume Sheet.

STATION	END AREA CUT SQ FT	END AREA FILL SQ FT	VOL OF CUT BCY	VOL OF FILL CCY	STRIP CUT BCY	STRIP FILL CCY	NET CUT BCY	ADJ CUT CCY	TOT FILL CCY	BALAN YARD. CCY	ALGEBRAIC SUM CCY	MASS ORDINATE CCY
2+00	0	324	0	952	0	181	0	0	1133	0	-1133	-1867
3+00	0	190										-3000
					✓	✓	✓	✓	✓	✓	✓	✓

Conversion Factor (BCY to CCY) = 0.90

You have 1 attempt left to complete this problem!
(* indicates method is okay, but the number is wrong.)

Checking your answers... (see above!)

Very Good! All of your answers are correct!

Use a Stripping Depth of 8 inches (= 0.6667 feet).

(Note: The end areas above are not to scale.)

NEXT to continue ◆ SHIFT-BACK for menu

Figure 4. Data and the solution of a problem presented by a table.

b. The second method is to choose randomly among several sets of data stored in the database. If the sketch of the problem statement is drawn according to the data, each student will get a different picture for the problem besides different numerical data. The problems in lessons dealing with shear force and bending moment diagrams are generally using this approach.

c. The third method is to generate new numerical data randomly within the limits indicated leaving the sketch of the problem unchanged.

The following is a list of computer techniques that proved to be valuable for lessons in engineering:

1. Flashing
2. Animation
3. Partial erase of the screen
4. Upscaling certain parts of graphs
5. Cursor usable by a student for reading the coordinates of the graphs presented in the display
6. Scrolling the problem data in a window-type arrangement done by a student when solving a problem on the screen
7. Movable boxes in tables (Figure 4)
8. Terminology and formulae database accessible by a key press
9. Simulation related to CAD
10. Combined CAI and CAD

PLATO lessons on road design and construction are under development at the Computer-Based Education Research Laboratory, University of Illinois at Urbana-Champaign. The project is sponsored by the United States Army Construction Engineering Research Laboratory, and the courseware developed is intended for the United States Army Engineer School at Ft. Belvoir. Some of the lessons have been used in a road design course in the Civil Engineering Department at the University of Illinois. Positive response from students and the faculty indicates that more PLATO lessons will be used in the course as soon as they are available. PLATO lessons in Statics are used in the School of Architecture and are being considered for usage in the Theoretical and Appied Mechanics Department, following their introduction at the University of Illinois Computer Fair, where they were demonstrated.

References

Kuznetsov, Helen (1986, February). Problem Solving Oriented CAI in University Statics Courses. 27th ADCIS Conference Proceedings, New Orleans, Louisiana.

Kuznetsov, Helen & Kao, Eugene (1985). Mass Diagram Analysis [Computer-based instructional program]. Urbana, IL: University of Illinois, CERL.

Kuznetsov, Helen & Kao, Eugene (1985). Mass Diagram Construction [Computer-based instructional program]. Urbana, IL: University of Illinois, CERL.

Kuznetsov, Helen & Kao, Eugene (1985). Earthwork Volume Calculations [Computer-based instructional program]. Urbana, IL: University of Illinois, CERL.

Kuznetsov, Helen & Kao, Eugene (1985). Mass Diagram and Equipment Utilization [Computer-based instructional program]. Urbana, IL: University of Illinois, CERL.

Kuznetsov, Helen, Sfondilias, John, Grisey, Matthew L., Williams, Odie (1985). Vertical Alinement [Computer-based instructional program]. Urbana, IL: University of Illinois, CERL.

Kuznetsov, Helen, Clapp, Elizabeth J., Davis, Dennis M., Felty, J. Michael (1985). Road Design [Computer-based instructional program]. Urbana, IL: University of Illinois, CERL.

Kuznetsov, Helen, Clapp, Elizabeth J., Felty, J. Michael (1985). Clearing, Grubbing and Stripping [Computer-based instructional program]. Urbana, IL: University of Illinois, CERL.

Contact: Helen Kuznetsov
University of Illinois
252 CERL
103 South Mathews Avenue
Urbana, IL 61801

217-333-3247

A STRATEGY FOR SIMULATIONS OF DISPUTE RESOLUTION PROCESS FOR USE IN LEGAL EDUCATION

M A K LAMBIRIS

Rhodes University

ABSTRACT

This paper describes and explains the structure of, and the theory behind, a series of four computer-assisted lessons which have been developed for use in the teaching of law. Produced on the PLATO® system, the lessons simulate dispute resolution processes and aim at replicating the conventional problem-solving exercises which are a traditional and important component of legal training.

INTRODUCTION.

Much preliminary work has been done on how a computer can be made to interpret and apply the law, but substantial difficulties remain to be overcome. One major obstacle is the lack of a universally accepted theory to explain the recognition, interpretation and application of the law. Another area of difficulty is the inexact linguistic structure of legal rules. These are but two of the difficulties that will make progress in this field slow. The question now addressed is whether, despite these difficulties, computer-assisted instruction can be useful in tertiary level legal education. Even if computers cannot yet be made to function as decision makers, can they be used to help students comprehend the law and develop the skills necessary to operate a legal system? It appearsthat this question can be answered in the affirmative. By using the conventional processes of legal education as models for computer-assisted lesson designs, many of the traditional objectives of legal education can now be accomplished in this new form. One such conventional process is legal problem-solving, the process whereby disputes are resolved by the application of rules of law. This paper describes an analysis of the conventional problem-solving process and the structure of a computer assisted lesson designed to simulate it.

Some clarification of terminology is necessary. Designating the lesson which will be described below as a simulation requires qualification. The simulation is of a system or process that defies precise scientific or mathematical analysis. The process simulated is not mechanical, formal or technical in nature, but appears rather to be of an informal intellectual kind. It is the process of apparent logic by which an appropriate answer to a specific question in terms of recognised legal rules is arrived at, often informally, sometimes even intuitively, with no obviously apparent structure or sequence. Nevertheless, the process can be analysed and broken down into identifiable parts. While it does not yet seem feasible to get a computer to perform these problem-solving tasks itself, it is possible for a computer to supervise a student problem-solver while he engages the problem. In a computer-assisted lesson, each step of the process can be simulated in order to engage the student in the same activities as the real life problem-solving situation would demand. Thus, even if computer scientists cannot yet produce a computerised decision maker, computers may still be able to help train human lawyers.

CONTEXT AND SITUATION

A procedural simulation requires a context. In the present case, the context within which the dispute resolution procedure is placed is the legal system and its various rules which will govern the decisions to be made. The simulation discussed below uses Roman Law as the context. This legal context provides the parameters within which decisions, classifications etc. are to be made. Contexts are of course alterable without the fundamental structure of the procedure necessarily being affected. Thus, as long as the fundamental processes behind legal reasoning remains similar, the validity of a simulation should be recognizable despite different legal contexts. The lesson design is therefore transferable to fields of law other than Roman Law.

As well as context, a simulation requires situation. This is provided by the particular facts of the individual case presented for resolution. Any dispute which requires resolution in terms of legal rules is an appropriate situation for a simulation of the kind to be described. The complexity of the facts and the extent to which the issues are hidden rather than obvious are among the variables that can be manipulated to increase or decrease the degree of difficulty of the simulation for the problem-solver.

AN ANALYSIS OF THE CONVENTIONAL PROCEDURE.

Since the analysis of the dispute resolution procedure on which the simulation described below is based may not be familiar to, or shared by, everybody, it may be helpful to give a brief outline of the different stages involved. The analysis is based on the deductive reasoning which is characteristic of the Romano-Germanic legal systems of the world. (Different stages in the process may be identified in, say, Common Law systems.) There appear to be at least seven distinctly identifiable stages in the process by which legal rules are applied to resolve a dispute. These stages may be summarised as follows:

1. The ascertainment, organisation, comprehension, sifting and evaluation of the available evidence (facts) which is provided by the parties to the dispute.
2. The perception, in broad terms, of the issues which are raised by the facts of the case.
3. The classification or categorization of the issues perceived in order to deduce the applicable legal norms.
4. The formulation of the issues in precise terms and at an appropriate degree of abstraction.
5. Retrieval of, research into and interpretation of the relevant law.
6. The application of the law to the facts of the case and the drawing of reasonable conclusions.
7. Deciding the case or predicting its likely outcome, that is to say, selecting the best of the possible solutions to the case.

These stages are not necessarily tackled in a strictly linnear order, and they are not accomplished in isolation from other background factors, which must be recognised. The process can be illustrated on a flowchart. (See below.) It may also be noted that the process involves a full range of the educational objectives listed by Bloom, both in the cognitive and affective domains, namely, knowledge, comprehension, application, analysis, synthesis, receiving, responding, valuing, organization and characterization.

Problem-solving is widely utilized in law schools, in order to develop these abilities in law students. A computer-assisted simulation must not loose sight of these objectives if it is to be judged acceptable in a law curriculum.

THE STRUCTURE OF THE SIMULATION.

In simulating the dispute resolution procedure, each of the stages identified above is used as the basis for an interactive, computer-assisted task. The first aim in designing each task is to exercise the student in as many of the usual intellectual skills as possible. Even within the limitations of current technology, questioning techniques are sufficiently flexible to avoid the more obvious disadvantages of multiple choice formats and thus reduce the element of guesswork or of pure recall responses. The second aim is to preserve the unity of the problem-solving process. The various stages of the process must remain closely interrelated and interdependent. Throughout the simulation, the decisions and choices that have to be made must relate to the particulars of the situation (that is, the facts of the case in question) and must be logically consistent with all of the choices made earlier in the exercise. Provided this unity is preserved in the lesson design, the process is likely to be a useful model in building and developing an independent problem-solving ability in the student which he will be able to transfer successfully to other environments.

The simulation which is described below is one of a series of four such lessons designed and implemented on the PLATO system at Rhodes University, Grahamstown, and used as a component in the teaching of Roman law to undergraduate and postgraduate students. Apart from the usual introductory passages and instructions, the simulation begins with a presentation of the facts of the case. This takes the form of statements made in the first person by the parties to the dispute, each of whom tells their version of events in turn. The student can return to these statements at will throughout the exercise, but since the facts are not on screen permanently, they must be stored by the student either in their own short term memory, or noted down in summary. Figure 1 is an example of a statement of facts. As in the conventional process, the student must be able to group related facts, and assess the materiality and significance of each fact. Computer assistance offers additional options which are not conventionally available, such as allowing the student to elicit further relevant information from the parties by asking questions for detail. This can be made manageable, (from the programming point of view) by allowing the student to select the most relevant questions from a list of possible enquiries, the lesson providing appropriate feedback in response to each choice. This can be a good test of a student's general background knowledge of the law, since it is in terms of such background knowledge that particular facts are known to be relevant and of importance. See figure 2 for an example of this process, and figure 3 for an example of the feedback given.

In the second stage of the simulation, the student is required to perceive the issues in broad terms. He is presented with various statements, each of which may appear superficially to be an issue raised by the facts, but some of which may be discounted because of discernable flaws, such as being merely subsidiary to the main issues, too narrow in their scope so that they do not encompass the real issues adequately, irrelevant to what the parties actually differ about, and so on. The student is not told how many of the statements are correct, and feedback is delayed until he has made all his choices. In order to locate the proper issues, the student must be able to discern exactly what it is that each litigant wants, separating these basic claims from the complexities of fact surrounding them, and looking beyond the uncertainties that the litigants themselves may express. See figure 4.

The third stage of the simulation involves the categorization or classification of the issues. Happily, because the range of possible classifications is fairly limited and predictable, student input can be accepted on an open-ended basis if desired. Otherwise, a sequence of questions may be used to make the student consider the matter of appropriate classification. Figure 5 and figure 6 illustrate these procedures. CAI has an advantage over the conventional teaching process in that it ensures that attention is paid to this important aspect of problem solving which students often omit to consider specifically. Categorization is not always easy when a case involves elements from different branches of the law, or when the dispute involves a combination of, say, real and personal rights and ought to be considered carefully.

Evaluating a student's efforts to formulate the issues with precision is difficult to handle on a computer. Conventionally, a student constructs the formulation of the issue in his own words. More than one form of words may be perfectly acceptable in defining a given issue. There is no mathematical exactness, but only a more flexible verbal or linguistic precision that is required. Further, issues can be acceptably abstracted at different levels, to widen or narrow them, which results in additional possible variations. The conclusion is that it is not really feasible to program suitable judging responses for all the likely acceptable inputs on an open-ended basis. The solution adopted here is illustrated in figure 7. A series of interchangeable phrases, arranged in groups, allows the student to synthesise a formulation of the issue which, even if not entirely his own, at least is not clearly before him in a complete form as an option. The student makes his choices of phrases, and is judged, in accordance with the characteristics which identify a well formulated issue. These include, for example, that an issue ought to be expressed in general rather than in specific terms, and that issues ought to be formulated so as not to unnecessarily widen the question that must be decided. As in the conventional process, the student's task is to generalize the factual situation, so as to exclude non-distinguishing facts from consideration, in order to formulate the issue as a question which, when answered, will provide a solution to the case. Figure 8 shows an example of feedback given in response to an attempted formulation.

The fifth stage of the simulation is where the student must locate the appropriate sources of law and do research in order to ascertain the relevant rules which ought to be applied to the case. The student is encouraged to refer to the usual sources of law, breaking from the lesson to do the work in a law library or wherever his information retrieval system takes him. References to sources can be given on screen. The simulation thus reinforces rather than replaces this aspect of a lawyer's task, and this is considered very desirable.

Testing the accuracy and depth of the student's research can be done in the way illustrated in figure 9. Since rules of law have no single valid expression, a student must be able to recognize statements which are a valid and appropriate expression of a legal principle or rule, and distinguish these from statements which are inaccurate, ambiguous, irrelevant or misleading. Thus the student is asked to indicate whether he considers a given statement to be accurate and relevant, accurate but irrelevant, or inaccurate. Each statement is shown on its own, one at a time, to reduce the possibility of guesswork. No help is given first time around, to encourage reliance on texts and case reports in finding the answer. The distinction between the statements can be made quite subtle so that the student is well exercised in his ability to distinguish between the variations introduced by small word changes in a text, or by word order. This is essential training for a lawyer, and efficiently accomplished and reinforced with computer assistance. Although in the simulation the student does not articulate the law for himself, the level of involvement in the particulars of the alternatives can be made suitably testing. The feedback offered for wrong choices can often help a student understand the finer points of a legal rule and how it may operate.

The process of applying the law to the facts, drawing conclusions and choosing the best solution as the outcome can be dealt with in different ways. Figure 10 illustrates the option of providing a passage from which key words are missing, which the student supplies by drawing appropriate conclusions. This method rather suggests a 'one solution only' philosophy and a better approach would be to have the student select the best of alternative solutions, with responses to indicate the advantages and disadvantages of each.

CONCLUSION.

The validity of the simulation must depend, in the final analysis, on the extent to which it successfully engages the student in the sort of processes and the intellectual activities that the real world situation does. It would appear that, provided sufficient care is taken in the construction of the questions and feedback, the simulation is capable of exercising a very acceptable range of intellectual skills and attaining a wide range of higher level educational objectives. Initial implementation shows that students find the simulations challenging and that they are prepared to spend substantial time and effort on them, returning voluntarily on occasion to re-run the lessons. Perhaps most pleasing is the quite widespread feeling among students that the lessons enhance their understanding of the law and how it works in practise. Such simulations appear therefore to be effective in tertiary level legal education.

FIGURE 1

The Process of Legal Problem-solving.

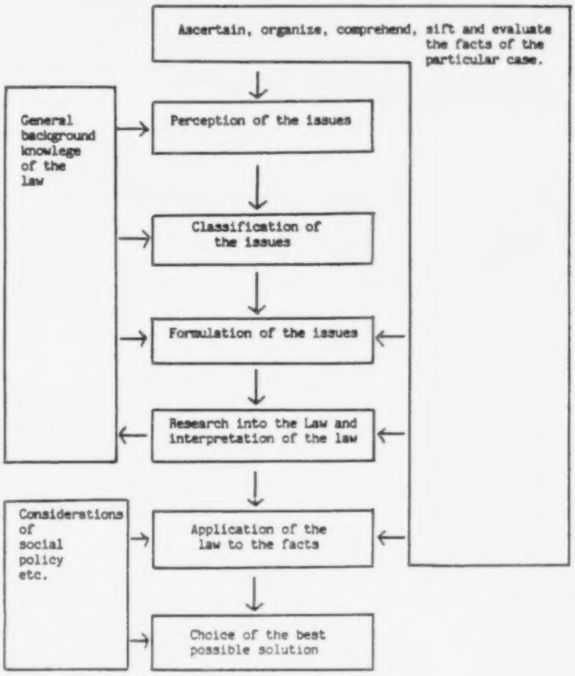
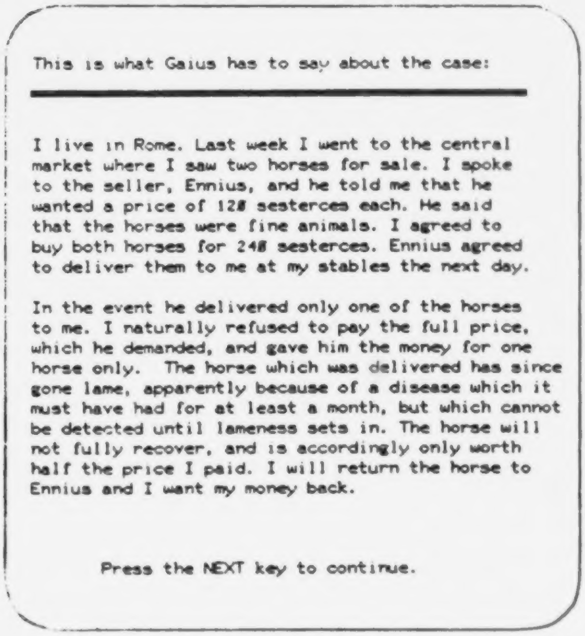

Ascertain, organize, comprehend, sift and evaluate the facts of the particular case.
General background knowlege of the law
Perception of the issues
Classification of the issues
Formulation of the issues
Research into the Law and interpretation of the law
Considerations of social policy etc.
Application of the law to the facts
Choice of the best possible solution

FIGURE 2

A Statement of Facts.



This is what Gaius has to say about the case:

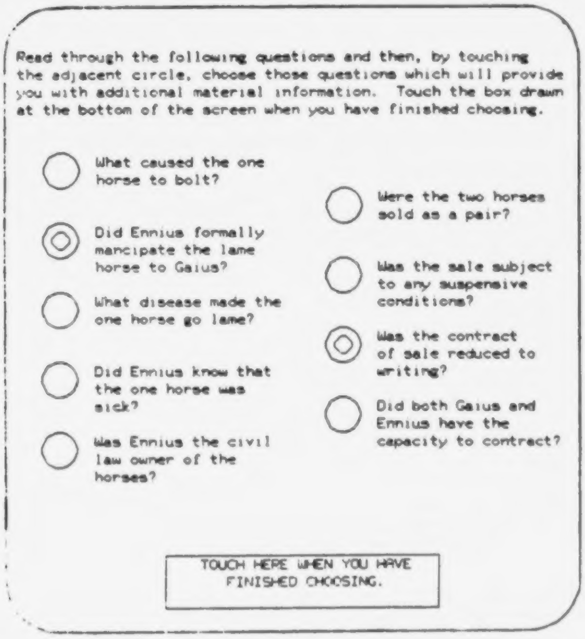
I live in Rome. Last week I went to the central market where I saw two horses for sale. I spoke to the seller, Ennius, and he told me that he wanted a price of 120 sesterces each. He said that the horses were fine animals. I agreed to buy both horses for 240 sesterces. Ennius agreed to deliver them to me at my stables the next day.

In the event he delivered only one of the horses to me. I naturally refused to pay the full price, which he demanded, and gave him the money for one horse only. The horse which was delivered has since gone lame, apparently because of a disease which it must have had for at least a month, but which cannot be detected until lameness sets in. The horse will not fully recover, and is accordingly only worth half the price I paid. I will return the horse to Ennius and I want my money back.

Press the NEXT key to continue.

FIGURE 3

Choice of Supplementary Questions.



Read through the following questions and then, by touching the adjacent circle, choose those questions which will provide you with additional material information. Touch the box drawn at the bottom of the screen when you have finished choosing.

○ What caused the one horse to bolt?

● Did Ennius formally mancipate the lame horse to Gaius?

○ What disease made the one horse go lame?

○ Did Ennius know that the one horse was sick?

○ Was Ennius the civil law owner of the horses?

○ Were the two horses sold as a pair?

○ Was the sale subject to any suspensive conditions?

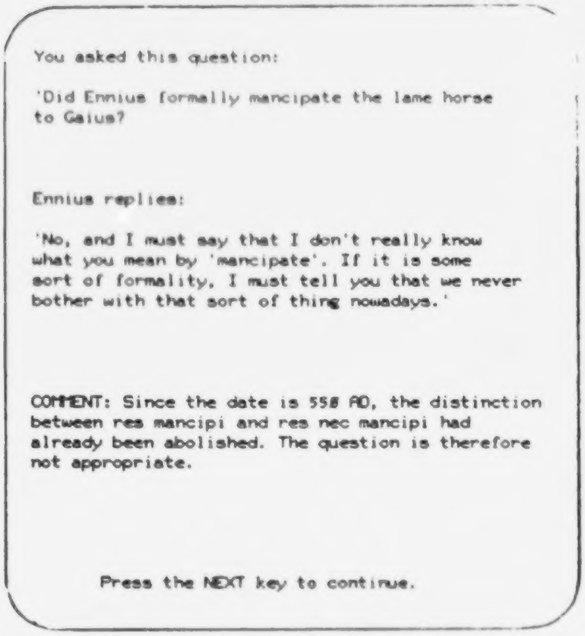
● Was the contract of sale reduced to writing?

○ Did both Gaius and Ennius have the capacity to contract?

TOUCH HERE WHEN YOU HAVE FINISHED CHOOSING.

FIGURE 4

Responses to Supplementary Questions.



You asked this question:

'Did Ennius formally mancipate the lame horse to Gaius?

Ennius replies:

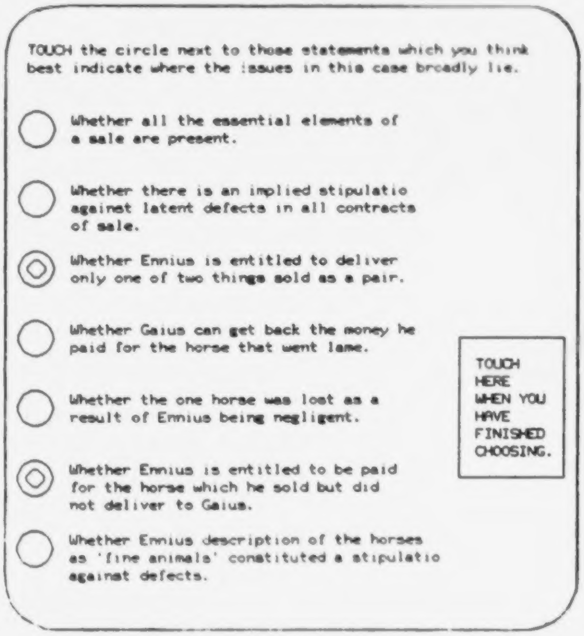
'No, and I must say that I don't really know what you mean by 'mancipate'. If it is some sort of formality, I must tell you that we never bother with that sort of thing nowadays.'

COMMENT: Since the date is 550 AD, the distinction between res mancipi and res nec mancipi had already been abolished. The question is therefore not appropriate.

Press the NEXT key to continue.

FIGURE 5

Perceiving the Issues Broadly.



TOUCH the circle next to those statements which you think best indicate where the issues in this case broadly lie.

○ Whether all the essential elements of a sale are present.

○ Whether there is an implied stipulatio against latent defects in all contracts of sale.

● Whether Ennius is entitled to deliver only one of two things sold as a pair.

○ Whether Gaius can get back the money he paid for the horse that went lame.

○ Whether the one horse was lost as a result of Ennius being negligent.

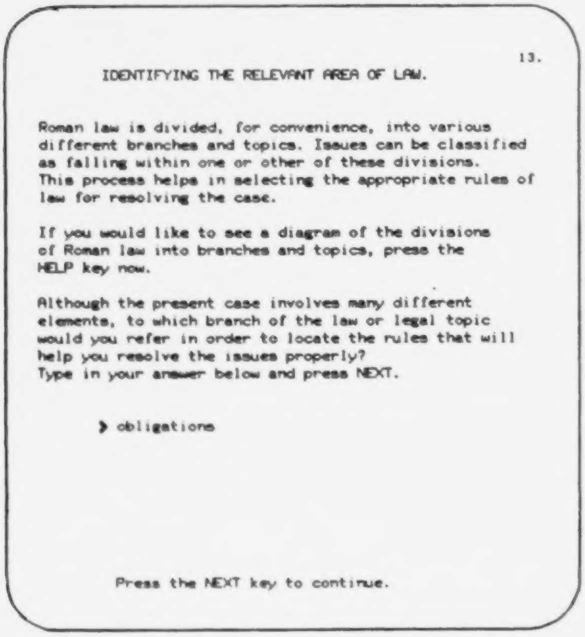
● Whether Ennius is entitled to be paid for the horse which he sold but did not deliver to Gaius.

○ Whether Ennius description of the horses as 'fine animals' constituted a stipulatio against defects.

TOUCH HERE WHEN YOU HAVE FINISHED CHOOSING.

FIGURE 6

Identifying the Area of Law Involved.



13.

IDENTIFYING THE RELEVANT AREA OF LAW.

Roman law is divided, for convenience, into various different branches and topics. Issues can be classified as falling within one or other of these divisions. This process helps in selecting the appropriate rules of law for resolving the case.

If you would like to see a diagram of the divisions of Roman law into branches and topics, press the HELP key now.

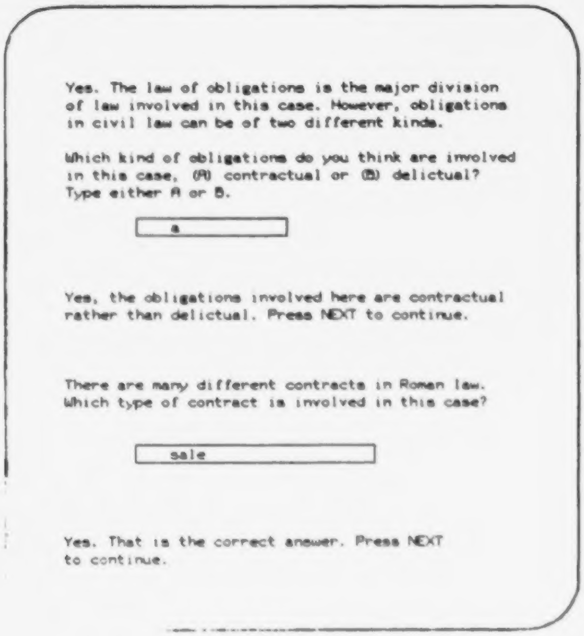
Although the present case involves many different elements, to which branch of the law or legal topic would you refer in order to locate the rules that will help you resolve the issues properly?
Type in your answer below and press NEXT.

> obligations

Press the NEXT key to continue.

FIGURE 7

Ascertaining the Branch of Law and Topic.



Yes. The law of obligations is the major division of law involved in this case. However, obligations in civil law can be of two different kinds.

Which kind of obligations do you think are involved in this case, (A) contractual or (B) delictual?
Type either A or B.

a

Yes, the obligations involved here are contractual rather than delictual. Press NEXT to continue.

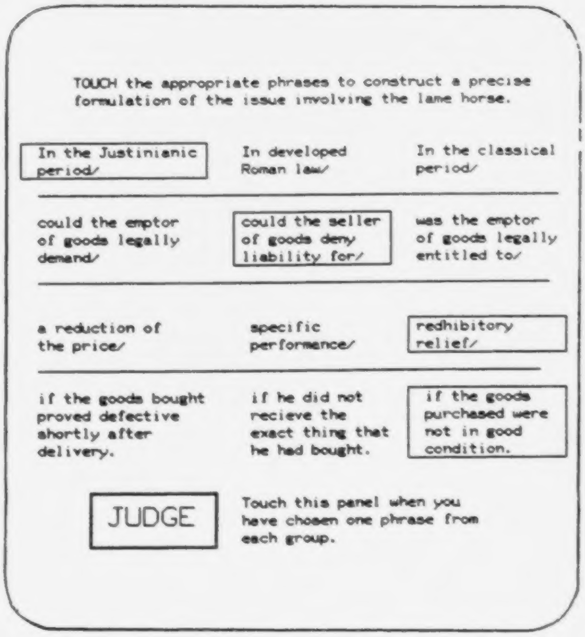
There are many different contracts in Roman law. Which type of contract is involved in this case?

sale

Yes. That is the correct answer. Press NEXT to continue.

FIGURE 8

Formulating the Issues.



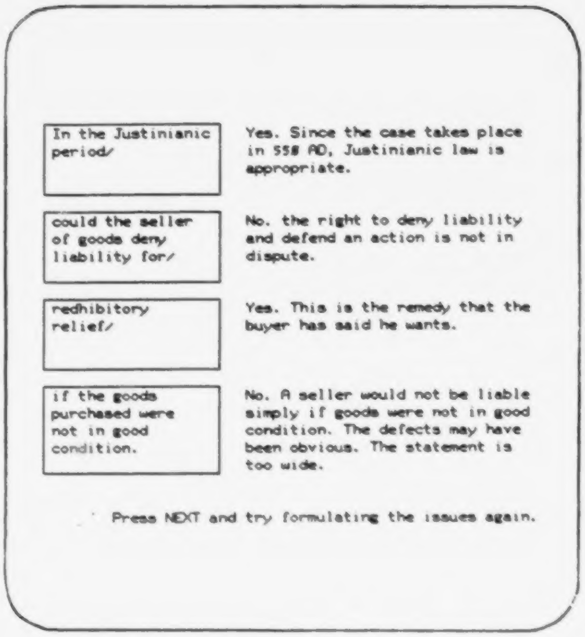
TOUCH the appropriate phrases to construct a precise formulation of the issue involving the lame horse.

In the Justinianic period/	In developed Roman law/	In the classical period/
could the emptor of goods legally demand/	could the seller of goods deny liability for/	was the emptor of goods legally entitled to/
a reduction of the price/	specific performance/	redhibitory relief/
if the goods bought proved defective shortly after delivery.	if he did not recieve the exact thing that he had bought.	if the goods purchased were not in good condition.

Touch this panel when you have chosen one phrase from each group.

FIGURE 9

Responses to an Attempted Formulation.

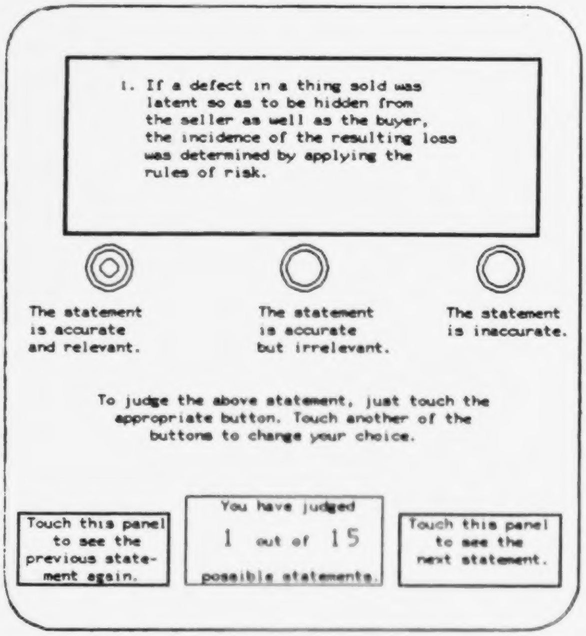


In the Justinianic period/	Yes. Since the case takes place in 550 AD, Justinianic law is appropriate.
could the seller of goods deny liability for/	No. the right to deny liability and defend an action is not in dispute.
redhibitory relief/	Yes. This is the remedy that the buyer has said he wants.
if the goods purchased were not in good condition.	No. A seller would not be liable simply if goods were not in good condition. The defects may have been obvious. The statement is too wide.

Press NEXT and try formulating the issues again.

FIGURE 10

Ascertaining the Relevant Law.



1. If a defect in a thing sold was latent so as to be hidden from the seller as well as the buyer, the incidence of the resulting loss was determined by applying the rules of risk.

◉ The statement is accurate and relevant.

○ The statement is accurate but irrelevant.

○ The statement is inaccurate.

To judge the above statement, just touch the appropriate button. Touch another of the buttons to change your choice.

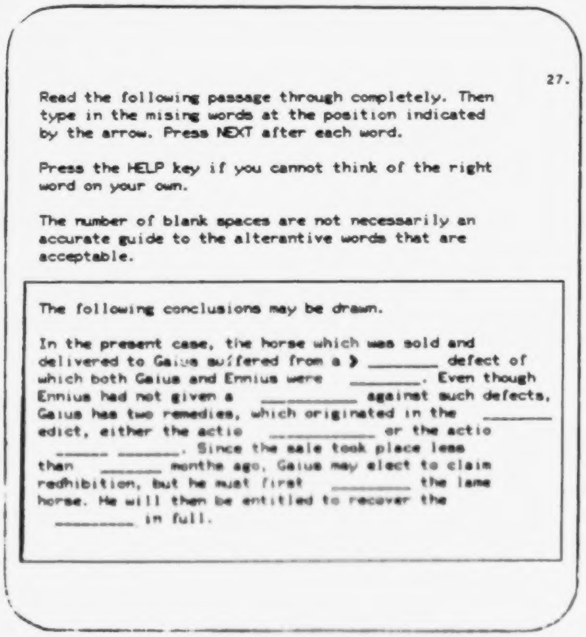
Touch this panel to see the previous statement again.

You have judged 1 out of 15 possible statements.

Touch this panel to see the next statement.

FIGURE 11

Application of the Law to the Case.



27.

Read the following passage through completely. Then type in the mising words at the position indicated by the arrow. Press NEXT after each word.

Press the HELP key if you cannot think of the right word on your own.

The number of blank spaces are not necessarily an accurate guide to the alterantive words that are acceptable.

The following conclusions may be drawn.

In the present case, the horse which was sold and delivered to Gaius suffered from a › ________ defect of which both Gaius and Ennius were ________. Even though Ennius had not given a ___________ against such defects, Gaius has two remedies, which originated in the ________ edict, either the actio ___________ or the actio ______ _______. Since the sale took place less than _______ months ago, Gaius may elect to claim redhibition, but he must first ________ the lame horse. He will then be entitled to recover the ________ in full.

PLATO is a registered trademark of Data Control Corporation.

BIBLIOGRAPHY.

S.R. Alessi and S.R. Trollip, COMPUTER BASED INSTRUCTION (1985) Prentice-Hall.

B.S. Bloom, J.T. Hastings and G.F. Madaus, HANDBOOK ON FORMATIVE AND SUMMATIVE EVALUATION OF STUDENT LEARNING (1971) McGraw-Hill.

J.A. Chambers and J.W. Sprecher, COMPUTER ASSISTED INSTRUCTION (1983) Prentice-Hall.

C. Ciampi, (Editor) ARTIFICIAL INTELLIGENCE AND LEGAL INFORMATION SYSTEMS (1982) North-Holland.

ACKNOWLEDGEMENT

I am indebted to G E Oberem, Head of the Computer-based Education Unit, Rhodes University, for his assistance and encouragement in developing these lessons.

Acknowledgement is also due to the Human Sciences Research Council and to the Attorneys, Notaries and Conveyancers Fidelity Guarantee Fund for the generous financial support given in attending this conference.

Contact: M. A. K. Lambiris
Rhodes University
P.O. Box 94
Grahamstown 6140
South Africa

0461 2-2023

ADCIS Proceedings, November 1986

Academic Computing Philosophy:

An Often Ignored Necessity

William W. Schnippert

Elmira College, Elmira, New York 14901

Abstract

Academic computing plan and policies that spring from an Academic Computing Philosophy are presented. Examples of an academic computing philosophy, computing plan, software purchasing guidelines, computer use, plagiarism, and copyright policies are given. Recommendations for creating "tailor made" philosophy, plans and policies are given.

INTRODUCTION The appearance of computers throughout the curriculum has dramatized the absence of a philosophy of academic computing and the accompanying plan and policies that guide us in the decisions that involve computers.
The fundamental statement that is needed is a statement of academic computing philosophy. From the academic computing philosophy and a knowledge of a college's resources, it is a relatively straight forward process to create an academic computing plan. In addition, the academic computing philosophy enables us to create policies concerning: academic computing facilities use, academic software purchase guidelines, academic dishonesty, and copyright protection.

Academic Computing Philosophy

COLLEGE MISSION:
Elmira College has an established tradition of a liberal arts college. "The mission of Elmira College is to offer its students a libera/professional education of sufficient breadth and depth to enable them to become more productive and responsible in contemporary society."* This mission implies our primary goal should be to develop transferable intellectua- and personal skills, aesthetic sensitivity, confidence, ability to make decisions, evolve personal standards and values, and encourage on-going personal and intellectual growth. This mission also implies that we should prepare our graduates for graduate school or a specific career as a secondary goal.
*Elmira College Bulletin

THE COMPUTER
The computer is a tool. Like all tools it becomes integrated into our society not just as an object but also as a metaphor. As we learn to use a tool, the tool changes how we view the world by both expanding and limiting our understanding of the possible. It is critical that we understand not only how the computer works and how to use it, but more importantly, how the power and limitations of the computer effects our society and our thinking.

ACADEMIC COMPUTING PHILOSOPHY:
The Academic Computing Philosophy of Elmira College is to prepare students and faculty to face and control the advent of the computer and information technologies, and at the same time use these technologies with a view to making education itself more effective, especially in terms of accessing and processing of information to assist students in gaining mastery in their chosen discipline. Knowing what the computer can do, how it works, and how to use the computer is important as a means of knowing for what the computer should be used. The "should of computer use" is the primary goal of academic computing, and implies

that our students understand use uses and abuses of computers so that they may make intelligent informed decisions concerning computers.
A major goal is to provide students with computer related perspectives and skills for a life enriched by computers rather than a life controlled by computers.
From this liberal arts perspective, the use of computers should enable students (and faculty) to expand the breadth of knowledge into the area of information science, to develop depth in critical and analytical capacities, and to study the impact of computers on man and culture. The pursuit of this goal should not be construed to mean we focus our curriculum on training data-entry personnel, computer operators, secretaries, programmers, business executives, engineers, or even computer scientists. These "how to use the computer" positions are by-products of trying to understand for what the computer should be used.

ACADEMIC COMPUTING PLAN

Philosophy of Academic Computing
Long Term Goals
Short Term Objectives

1. Facilities: Hardware, Software, work space
2. Personnel: Administration, Faculty, Staff
3. Budget: Process, distribution
4. Community: Environment, expectations

ACADEMIC COMPUTING POLICIES

Academic Computing Software Purchase Guidelines

The real question is "What are we trying to accomplish with the software and the computer that runs it?" Improve teaching? Support Faculty research? Support Faculty Development? Support student development?
GENERAL APPROPRIATE USE

1. Is the software of a general nature and usable by many faculty/students
-or 2. Is the software course specific and usable by only a few faculty? If so, can the software be justified because it is pedagogical superior, or necessary to reach the objectives of the class?
-or 3. Can the software be justified on the basis of promoting computer education or computer interest among the faculty? This committee generally does not approve software requests to support research
-or 4. Can the software be justified as an evaluation copy? Time limits?
BALANCED USE OF SOFTWARE BUDGET

1. Is the software appropriately priced? For multiple copies?

2. Is the software an appropriate use of computers? If not can the package be justified as a bad example?

3. Is the cost of the requested software proportional to the expected usage? If not can it be justified on the basis of the kind of software it is or the idea the software embodies?

SPECIFIC REQUEST:

1. Information for requested software: description, cost, distributor/producer, copyright status, documentation, support, and policy for multiple copies, back-up, replacement, up-dates.
2. Tried out by faculty members and some of the intended students?
3. Purpose for purchase of the software.
4. Sufficient number of copies for the intended purpose?
5. Ethical considerations. Agreement with Elmira College academic computing philosophy and objectives.

USE OF ACADEMIC COMPUTING FACILITIES

The Computer Laboratory Assistants will enforce the following priorities for student use of the Computer Laboratory.

1. Reserved use for classroom demonstrations.
2. Use by Elmira College students for classroom assignments
3. Use by Elmira College students for non-class activities.

ACADEMIC COMPUTING DISHONESTY:

All acts of academic dishonesty are unethical and may result in college imposed penalties (ie: denied access to the computer center, expulsion from school, etc.) Under certain circumstances, academic dishonesty may result in prosecution under Federal, State, and Local laws. Computer Center personnel will follow and enforce the following guidelines.

Definition and examples of academic dishonesty:

Academic Dishonesty is any treatment of, or representation of work as if one were fully responsible for it, when it is in fact the work of another, or work in which one has received unacknowledged assistance from others.

Academic Dishonesty includes, but is not limited to:

1. Intentionally submitting any fraudulent or plagiarized academic work. Examples include, but are not limited to:

A. Verbatim use of a quote without quotation marks.
B. The use of another person's idea or work as one's own without acknowledging its use.
C. Submission of a paper prepared by another person as one's own work.
2. Giving or receiving answers and/or using any materials pertinent to the examination without the permission of the instructor.
3. Stealing, manipulating or interfacing with any academic work of another student.*

*Elmira College Policies & Procedures Handbook, pg. 37.

Additional computer specific acts of plagiarism include, but are not limited to the following:
1. Copying a discarded printout.
2. Transferring programs, text, or data from another person's account or storage medium to your account or storage medium.
3. Copying another person's assignment or exam from a video screen.
4. Asking a consultant or anyone else experienced in computing how to design a program or complete an assignment. A consultant or supervisor can help you with the syntax of a programming language, the meaning of an error message, and with questions on how to use the operating system and utilities (ie: the DEC EDT text editor). If question can be answered without knowing the details of an assignment, then it is probably legitimate.
If you are unsure about an assignment, contact your instructor.
5. All acts of alleged academic dishonesty will be reported following the guidelines outlined on pages 37-40 of the Elmira College Policies & Procedures Handbook.

Copyright Protection includes both protecting the author from the unethical user and protecting the author and the institution from each other.

Copyright laws and licence agreements.

The majority of Software purchased by Elmira College is copyrighted and cannot be copied by someone who did not purchase the programs or sign the license agreements. All software is either copyright protected or is in the public domain.

Public Domain software is not copyright protected and may be copied and/or modified by anyone.

Copyright protected software means that the author(s) has expressly stated that his work is original and copyrighted (this can be done simply by stating in writing that the author retains all copyrights). This gives the author(s) specific rights and protection by federal law.

Background for Copyright Law.

The original copyright laws were written before photocopiers and computers wer widely available. Since technology has made quick and easy copies of many items available, to nearly everyone, the federal copyright laws have been revised. The revisions of the copyright laws focus on the problem of number of copies necessary for legitimate use balanced with adequate protection against financial loss for the author(s).

In terms of copyrighted computer software this usually means that copies of the original software can be made for use by the purchaser. Specifically, this means copies made for archival (backup) purposes and copies made when the software is copied into RAM memory for use by the purchaser are legitimate. Multiple copies for the owners use on different disks, that are not used at the same time (concurrently), are usually considered legitimate. Although the original may be sold, loaned, or given away (first sale doctrine), copies may not be made to give away or for sale or for loan, while the original owner continues to use the software. Copies made for these purposes (creation of multiple users) violate federal copyright law (a felony) and may also violate state and local laws.

A current situation that causes some confusion is the fact that some authors distribute their software with some form of "protection" in addition to the protection provided by a copyright statement, and some authors protect their software only with the copyright statement. Many people incorrectly interpret the lack of extra "protection" by some software authors to mean that "copy for distribution" is permitted. This interpretation is wrong. All copyrighted software is protected by the same federal copyright laws. Failure of an author(s) to provide extra "protection" for his copyrighted software does not give permission to "copy for distribution".

A second situation that also causes confusion is the creation of a third category of copyright protection called "shareware" by some authors. These authors give users of their software specific permission to do certain kinds of "copy for distribution" activities. These "copy for distribution" activities are often "copy to give away" or "copy to try it out" rights. Since these rights obviously vary from author to author it is not possible to give any general rules except to read the "author given rights" very carefully.

The computer center will follow and enforce the following guidelines in regards to computer software.

1. You may not duplicate using Elmira College computers any copyrights software that does not have author permitted "copy for distribution" rights, which has not been purchased by you. The computer center labels their copyrighted and copy prohibited software as: PROTECTED BY FEDERAL COPYRIGHT LAW - CANNOT BE COPIED.
2. You may not duplicate using Elmira College computers any copyrighted software that does have author permitted "copy for distribution" rights, except for those specific "copy for distribution" rights defined by the author. The computer center labels their copyrighted software with author defined copy permission as: PROTECTED BY FEDERAL COPYRIGHT LAW - SPECIFIC AND LIMITED COPY PERMISSION GRANTED BY THE AUTHOR - READ THE COPYRIGHT STATEMENT CAREFULLY.
3. If you intend to use a purchased or borrowed program for your coursework on the computers belonging to Elmira College, you are required to bring documentation of proof of purchase and documentation for use.
4. Elmira College owned software and documentation is kept on reserve in the software library for use by members of the college community. Students, faculty, staff, and administrators may check out the software, manuals, etc. for use at the computer center, or for use in special permission projects.
5. Please keep any data, files, or programs created by yourself on your own disks, not on the software library disks. Private information on Elmira College disks cannot be protected against invasion of privacy, and copyright protected material cannot be protected against piracy.
6. Creation of copyright protected software on Elmira College computers and/or using Elmira College software, may give Elmira College specific rights to and claims on the software (works made for hire doctrine). All would-be software authors should use their own computer systems on their own time, or have a written agreement with Elmira College concerning the created software.

RECOMMENDATIONS:

Creating the academic computing philosophy and the corresponding plans and policies requires the broadest possible perspective and is best done through a joint (faculty, administration, staff, and student) committee structure.

Charges for a committee on Academic Computer Resources:

1. Academic computing policies.
    a. Academic computing philosophy statement.
    b. Academic computing plan.
    c. Computer literacy plan.
2. Allocation of computer resources.
    a. Review requests for the purchase of shared academic software.
    b. Make recommendations for the purchase of computer hardware.
    c. Monitor computer usage to assure the fair use of resources.
    d. Set procedures to honor copyrights, ethical statements, etc.
3. Personnel
    a. Academic computing support staff recommendations.
    b. Review job descriptions for academic computing support staff.

Contact: William W. Schnipper
Box 7019
Elmira College
Elmira, NY 14901
607-734-3911, ext. 323

ADCIS Proceedings, November 1986

INTEGRATION OF COMPUTER BASED INSTRUCTION INTO EDMONTON POLICE DEPARTMENT TRAINING PROGRAM: A CASE STUDY IN PRODUCTIVITY

William A. Wosar, Edmonton Police Department, Training Section
Michael Szabo, Ph. D., The University of Alberta

ABSTRACT

Law enforcement places great demands on the discretionary abilities of peace officers, plus equal demands on the effectiveness of police department training programs. In cases of negligence, civil action may be taken against the department; training is the inevitable first target of scrutiny. The type of training received, required standards of proficiency, frequency of testing and performance results will invariably be questioned in court. When all personnel receive well-researched and standard training materials and demonstrate objectively verifiable understanding, documentation of training is easier to maintain and more effective in cases of civil litigation (Higginbotham, 1980). The challenge, of course, is to provide standard, well documented training in the face of budgetary realities.

In 1973, Ivor Davies declared in *Competency Based Learning: Technology Management, and Design* that society can no longer afford the high expense and low productivity of the nation's most costly undertaking - education. Thirteen years later with severe pressures to cut back at every level of government, this concern is even more valid, and particularly challenging to municipal law enforcement agencies who must seek methods to ensure efficiency and effectiveness. Higginbotham (1980) indicated studies continually confirm that improved and continual officer training upgrades police performance, which in turn leads to increased productivity.

Fortunately, the development of Computer Based Training (CBT) has provided, at ultimately reduced costs, highly consistent training with improved control of material (Kearsley, 1984). With CBT, a training section can document in detail the performance of each trainee on every training objective. The benefits of CBT have been well documented, while in recent years the costs have dropped dramatically. In August 1983 the Edmonton Police Department (E.P.D.) implemented CBT and, as the record of its integration, development and results attests, the innovation has proven highly successful.

CONVENTIONAL INSTRUCTION

The E.P.D. Training Section, long recognized as a leader in its field, began with classroom instruction, moved to roll-call training in the mid-1970's and switched to CBT in 1983. The conventional form of training used by the E.P.D. until the mid-1970's consisted of a two-week refresher course given to all members once every five years. All members were required to attend 80 hours of conventional classroom instruction. The difficulty in disseminating in two weeks new policies, procedures and changes to law in the preceding five years proved awkward at best. Class size (50), non-motivating lectures, six man-years of required instructor time and inadequate documentation of performance added to the inefficiency of the refresher courses. As Kenneth Hultman notes, behavior demonstrated by some of the participants inevitably interferes with the ability of others to effectively achieve workshop objectives. Some problems of the conventional method are: arriving late; missing sessions; clowning; dominating discussion and passivity (Hultman, 1982).

Roll-call training, originally instituted in major centres in the United

States, was investigated by the E.P.D. and found to be suitable for implementation. Audio-visual presentation during roll-call, prior to tour of duty, was viewed as a viable instructional supplement. Roll-call training was quickly and easily adopted and hastened the exit of the existing in-service training.

A field study within the Department prioritized the roll-call topics to be developed. Implementation of in-house productions was widely accepted, with minor token resistance. Production itself was not without problems. Too often last minute changes to content prior to distribution were dictated by those involved in the review process. The Instructional Television Unit was born out of the need for a responsible research and development team.

Evidence surfaced that the message to be delivered with the aid of video during roll-call was being altered all too often. Elements contributing to this problem were interpretation by the instructor; scheduling due to shifts; lack of post-session discussion; manual record-keeping; and in particular, as in the case of the two week refresher course, NO measure of achievement or retention. The loss of manpower from the street was seen as the most undesirable aspect of all. Once again an alternative had to be found.

IMPLEMENTATION OF CBT

This replacement for roll-call training would have to increase the learner's retention factor and motivation to achieve and be utilized 24 hours per day, 7 days per week. To reduce both training time and cost, the training had to be made available at any time. The training had to be controlled, standardized, time efficient and cost effective. Further, it must be Decentralized, Individualized, and In-service Training (D.I.I.T.). Most important there had to be extensive record-keeping and a measure of retention.

CBT was familiar to certain Training Section personnel. The benefits of research and developments in the field of CBT were compiled and presented to the executive as a proposal to develop CBT within the Department.

Consultation with qualified personnel at the University of Alberta led project leaders to the PLATO system. The lack of existing courseware in the field of law enforcement required local custom courseware development. The ability of PLATO to schedule, deliver, diagnose and record Computer Managed Training (CMT) was well suited to the needs of the E.P.D. Training Section. In 1983, all modules were converted to PLATO's CMT system. In 1984, the University made available its programmerless authoring system (OMNISIM) which was used to create computer assisted training tutorial lessons. Best of all, both systems can be used by experienced instructors who have no programming skills. Funding was approved on the recommendation of the executive who saw PLATO system applications as cost effective.

DEVELOPMENT

Priorities in course module topics were established by conducting a survey throughout the Department. The Learning Resources Unit, formerly the Instructional Television Unit, is responsible for the research, development and validation of each CBT module.

The unit comprises one Sergeant (subject matter expert and resource co-ordinator); a Computer Based Training Specialist; and a Video Resource Specialist.

Members of the E.P.D. have been required to achieve one hundred percent mastery on 32 modules developed since August 1983. The modules fall into two categories:

- CRIMINAL LAW

  Powers of Arrest **
  Powers of Search **
  Child Abuse
  Young Offenders Act
  Operate While Impaired *
  Criminal Driving Offences *
  Domestic Disputes I & II
  Elements of Crime
  Sexual Assaults
  Appearance Notices
  Fraud Offences
  Weapons Offences I & II
  Public Order Offences *
  Laws of Evidence
  Blood Warrants

- SPECIALIZED INVESTIGATIONS

  High Risk Incidents
  Vehicle Thefts
  Suspect Identification
  High Speed Pursuit
  Dog Unit Services
  Vehicle Seizures
  Bomb Threat Calls
  Interviewing & Interrogation

** Mastery required each year
* Presented twice due to Criminal Code amendments

D.I.I.T. lessons have three components: tutorial Computer Assisted Training (CAT), videotape, and computer managed training.

The support of management is required if any program, particularly an innovative one, is to succeed. Therefore managers within the Department, at the rank of Superintendent, are surveyed on a monthly basis. They are briefed on why this module of training has been developed and asked for comments pertaining to its structure and content. Suggestions for future topics of interest and need are solicited at the same time.

D.I.I.T. was initially intended to serve the needs of operational (street) personnel. The front-line ranks are most deserving of excellent training. In view of their many and varied daily encounters, it would be inappropriate not to allow them the opportunity to state their views on training materials. Their experiences shed light on the problems that may be solved by CBT in the future. Electronic messaging provides for viewpoints, queries and replies to be made at any time during a terminal session. Learning Resources Staff see the expressed views of 1100 students as a most powerful form of validation. More than 1,000 student comments were logged in less than one year.

RESULTS

Discussions about CBT usually centre on cost benefits and what it can do to improve job performance.

Cost Benefits

To effectively compare the costs of two forms of training, even though one has been replaced, they must be calculated at present costs.

The savings realized by the on-going use of CBT versus the conventional classroom refresher once every five years for two weeks is outlined in Table 1 below.

The left column indicates the training costs for a two week refresher course, once every five years for a force of 1100 persons. Included is the time lost from operations over a two week period; the salaries of six instructors doubling as developers and instructors; secretary; consultants and the cost of materials.

The right column represents the costs of CBT at eleven (11) sessions per year for five years for 1100 men. Each monthly session requires a maximum of one(1) hour. The salaries of the Learning Resources Unit personnel totals $100,000 per year. The PLATO system is the mode of delivery.

Table 1
Cost Savings from Conventional to Computer Based Training Over 5 Years

	Conventional training	Computer based training
1. Frequency	2 weeks in 5 years	Monthly for 11 months
2. Time lost in training	88,000 person-hours	60,500 person-hours
3. Trainee salary costs ($25/hour)	$2,200,000	$1,512,500
4. Development costs (staff & materials)	$ 60,000	$ 500,000
5. Delivery costs	$ 270,000	$ 175,000
Total (3+4+5)	$2,530,000	$2,187,500

Net Savings = $342,500/5 years = $68,500/year

Several qualitative differences between conventional and computer based training, deemed important by E.P.D. are listed in Table 2.

Table 2
Qualitative Differences Between Conventional and Computer Based Training

Conventional training	Computer based training
• class size limit	• no classrooms
• limited interaction	• highly interactive
• inconsistent delivery	• standard and consistent
• boring lectures	• instructors specialized
• manual, limited performance records	• extensive, automatic performance records
• inadequate performance documentation	• competency/mastery based
• no measure of retention	• measures of retention

Improved Job Performance

If a discrepancy exists between actual job performance and requirements, then production, efficiency and eventually morale will suffer. An analysis of facts, tasks and behavior is required. The discussion of improved job performance will focus on the results generated by one module from each of the CBT lesson categories previously mentioned.

Specialized Investigations

In response to concerns expressed by the E.P.D. Dog Unit a module was developed in co-ordination with senior dog handlers. Scenes were too frequently "contaminated" by initial response units prior to the arrival of the dog and handler's arrival. The procedures to follow are relatively simple but were continually neglected, causing a track or search to end abruptly. The following results indicate that the intended message was received.

Table 3 indicates the average numbers of scenes attended and successful apprehensions with dog units before and after training was instituted. The percent change from the preceding 5 months was 27% in scenes attended and 45% in successful apprehensions.

Table 3
Improved Performance Related to Dog Unit Services Module

	Scenes attended	Successful Apprehensions
Before training (5 months)	*Mx* = 304/mo.	*Mx* = 15.4/mo.
Post training (7 months)	*Mx* = 385/mo.	*Mx* = 22.4/mo.
Percent change	+27%	+45%

The significant increase in apprehensions represents more than arrests at the scene of a crime, recovery of property and individuals sought by warrant. The apprehensions by the unit signify a significant reduction in man hours spent in following up unsolved incidents, tracing stolen property and seeking individuals at large.

Criminal Law

Peace officers are continually under pressure to provide service in situations that demand priority. The issuing of Appearance Notices is meant to aid in this endeavor. When an offence has been committed and the officer has secured the evidence; is satisfied with the accused's identification; is satisfied the offence will not continue; and is assured the accused will appear in court, the conventional arrest is not made and an Appearance Notice is issued. The Criminal Code dictates the offences which may and may not be handled by Appearance Notices. The time normally taken to transfer the accused to a detention unit may now be given to waiting response requests.

However, when an Appearance Notice is issued inappropriately or not written in accordance with recognized procedures it creates additional and unnecessary work. If any of the aforementioned prior requirements to issuance of an Appearance Notice are, in the opinion of the officer, suspect, then an arrest should be made. Inaccuracies in reference to date, time and place of identification processing and /or court hearing; crossouts; writeovers and spelling errors will sometimes void the issuance altogether. Identification processing is the logical consequence of an arrest but when an Appearance Notice is issued to a suspect individual there is a likelihood of the accused not appearing for processing or court hearing. When obvious mistakes are made in completing an Appearance Notice there is again a chance of the accused not appearing for processing or court hearing.

Both situations require that a Warrant Request 133 (5) C.C. be made for the accused. The following statistics outline the continuous rise for Warrant Requests in the last three months of 1984 versus the same period in 1983. The CBT module developed to alleviate the discretionary and clerical errors was delivered in January 1985. The remainder of 1985 is compared to the same period in 1984 in Table 4.

Table 4
Improved Performance in Reducing the Number of Warrant Requests

	Before training		
	1983	1984	change
October	157	198	+41
November	164	203	+39
December	156	178	+22
Mx =			+34

(cont'd on next page)

	After training		
	1984	1985	change
January	162	179	+17
February	182	128	-54
March	175	140	-35
April	216	148	-68
May	209	144	-65
June	218	103	-115
July	174	110	-64
August	174	121	-53
September	203	235	+32
October	198	157	-41
November	203	137	-66
December	178	92	-86
Mx =			-49.8

In the eleven months following the CBT lesson, 598 fewer Warrant Requests were made than the year before. This is a 26% decrease in the initial paper work alone. A Warrant Request when broken down into its components takes approximately an hour to process. This translates into a saving of 598 hours or 4 MAN MONTHS, in addition to the incalcuable time spent in tracing, arresting and processing individuals sought by the Warrant Requests.

D.I.I.T. has generated some unforeseen results since its inception. Constables with three years of service qualify for promotional exams. The content consists primarily of criminal law and its effective enforcement. These classes are in effect a form of upgrading. Criminal law basics have been addressed during recruit training. Gernazick, Lanoza and Nolan (1980) cite two characteristics emerging from their research as appearing instrumental in successful remediation: self-paced instruction and computer instruction. Table 5 below details the decrease in failures among the participants and the increase in the average score achieved after the introduction of CBT and its utilization as a review tool.

Table 5
Improved Performance in Constables' Promotional Examinations

	Before training		
	No. writing	No. failures	Ave. score
1981	81	14	68.5
1982	84	5	70.4
1983	86	2	73.5

	After training		
	No. writing	No. failures	Ave. score
1984	75	1	75.2
1985	68	0	76.6
1986	80	0	79.6

The absence of a failure during the last two years and steady increase in the average score of the participants lends support to the research of Gernazick, et al.

SUMMARY AND CONCLUSION

Several factors associated with the successful introduction and implementation of CBT in the E.P.D. can be identified. Many of these have already appeared in the literature.

First, there was clear dissatisfaction with previous forms of training, each of which had serious shortcomings. Next, Training Section staff were willing to become sufficiently knowledgeable and to investigate a CBT solution and to recognize its strengths and limitations.

A third factor is that the middle level management worked pointedly toward informing and educating senior management in understanding the potential and limitations of CBT as applied to their training requirements. Their success can be judged by the fact that the senior officer required all non-civilian staff to have hands-on experience with the training modules early on in the project.

Another major factor is the relative ease of development of training modules (courseware) using PLATO. Computer managed and assisted training modules were developed using PLATO Learning Management and OMNISIM. Both are sophisticated state of the art software packages which require no programming skills to use. Thus the Department was able to develop substantial amounts of custom courseware without resorting to expenses associated with training or hiring programmers.

Association with a major university was also beneficial. The E.P.D. benefitted in the areas of technical advice, validation, evaluation and a marketing outlet for its training modules. The university benefitted from increased experience in technology transfer and research opportunities for graduate students.

REFERENCES

Davies, I.K. (1973). *Competency based learning: Technology management and decision*. New York: McGraw-Hill.

Gernazick, R., Lanoza, C., & Nolan, R.D. (1982). Microcomputer remedial instruction. *The Computing Teacher, 10(2)*, 50-52.

Higginbotham, C.E. (1980). In-service training. *The Police Chief, 47*, 16-19.

Hultman, K.E. (1982). The trainer as scapegoat. *Training and Development Journal, 36*, 44-53.

Kearsley, G. (1984). *Computer based training: A guide to selection and implementation*. Don Mills, ONT: Addison-Wesley.

Contact: Michael Szabo
Department of Computing Services
302 Students' Union Building
The University of Alberta
Edmonton, Alberta T6G 2J7
Canada

ADCIS Proceedings, November 1986

The Development of a Total Curriculum for Physical Science Using Interactive Videodisc Technology

Stephen F. Brown, Ph. D.

Texas Learning Technology Group

ABSTRACT

The Texas Learning Technology Group, in conjunction with the Texas Association of School Boards, the National Science Center for Communications and Electronics Foundation, 11 Texas school districts, and vendors in the field of education is in the development and production phase of a total technology based curriculum for physical science. This paper details the problem of science education and its trend, the massive organizational efforts, the configurations of the classroom, and the design specifications. An appeal is made for greater interaction between the private sector and school districts in order to bring about the needed improvements in education.

INTRODUCTION

This paper presents an account of a major effort both at the Texas and the national levels to develop a complete technology-based curriculum. The technology is being used not only to solve a student learning problem, but to address a crisis in science teaching and to improve the methodology of processing student learning data in the classroom, the district, and eventually the state.

In the early portion of 1985, at a national conference devoted to discussions of the problems of education, two people met and discussed taking action to remedy some of the problems. The Executive Director of the Texas Association of Boards and the Chief Executive Officer of the National Science Center for Communications and Electronics Foundation, Inc. (NSCCEF) discussed how to effect a change in the educational process.

NSCCEF was founded to improve the quality of science and mathematics education, and has a mission of funding projects which use technology to address these areas. The offer was made to Texas that NSCCEF would provide $500,000 to be matched 3:1 by school districts willing to invest in an R&D effort.

TLTG quickly learned that physical science, a course normally taught at the ninth grade level, was deficient in teacher preparedness, student achievement, and student motivation. According to the National Science Teachers Association survey (NSTA, 1982), 55 per cent of the new science teachers in secondary subjects were certified or eligible for certification in 1981. This figure has decreased during the last five years. During the 1982-3 school year, about 18,000 mathematics and science teachers quit their teaching positions. About 37 per cent of these left for employment outside of teaching. According to the National Center for Educational Statistics (1983), the United States produced less than 1400 bachelor's degrees in the fields of mathematics and science education combined. Obviously, there is a severe need for technology to both educate the non-certified science teachers and to deliver quality instruction of the core concepts of the science. The salary differential between industry and education will widen in the next 10 years to a point where the shortage of certified teachers may lead to the nation losing its technological economic base. Taking this data and interviews with Texas school district administrators in to account, it was determined that physical science would be the subject of the first course.

The Texas Learning Technology Group (TLTG) was begun as a non-profit organization on September 30, 1985 with a two million dollar budget. One of TLTG's high level goals is to develop a 50 million dollar rotating trust fund so that additional courses can be developed. This fund could be supported by grants from foundations (this process has already begun) or by sales of the courseware, either to the districts, or to the state. Texas is one of five states which have a state textbook adoption process. It has been indicated that the state would be willing to buy the courseware as a de facto textbook for distribution to schools if it could be shown that the courseware can stand alone as well as a textbook.

DESIGN DOCUMENT

From October, 1985 through February, 1986, TLTG contracted with a vendor to develop a design document for the course. The development of this design document involved input from many facets of the educational environment. Four committees were formed to make recommendations. These committees were the technical, curriculum, evaluation, and teacher training committees, comprised of district administrative staff, teachers, university colleges of education faculty, designers, and producers. The busiest committee was the curriculum committee, which met for a total of approximately 70 hours. The problem which had to be overcome was the fact that although Texas has a set of essential elements, each school district approached interpretation differently. Curriculum guides in the districts ranged from 3 pages to a 15 pound box. Teachers and students were interviewed by course designers and a producer. In determining the instructional approaches to take, the interviewers looked for motivations in the lives of the students. The two most dominant themes were 1.) animated characters and 2.) scripts involving scenes from everyday life.

The technical committee was the stage for several debates between staunch Apple computer supporters and MS-DOS proponents. The key question regarding the Apples was "Do we need to consider the installed base ?" The final answer was NO. Districts with a large installed base of one type of computer were found to be utilizing the computers effectively for their original purpose, such as computer literacy. MS-DOS and IBM compatibility at the PC level was then agreed upon as the basic standard. This effectively opens up the market so that the districts benefit from the competition.

Level III interactive videodisc was chosen for its individualized instructional abilities. Digital or synthetic speech was chosen so as to allow for easier updates to content.

Since the cost of a classroom of interactive videodisc systems would be prohibitive to the 900 out of 1000 school districts making up 10% of the Texas school children, three configurations of the classroom were designed. Following is a description:

TEACHER STATION -- A single interactive videodisc system at the teacher's desk is used for providing the teacher with an electronic gradebook, a test bank, computer compatible scanner, word processing, and a course management system. A 25" monitor is used with the system to lead the class in discussions, content instruction, laboratory preparation, and activities (large screen projector is optional). The teacher should also become more competent with the content through separate teacher sections of the course.

TEACHER/GROUP STATIONS -- In addition to the teacher station described above, five group stations (for a 30-student class) may be added. With four or five students participating in activities and exercises together, advantage can be taken of the peer and social learning aspects of instruction. These stations each consist of an interactive videodisc system with a 19" monitor.

FULL CLASSROOM -- This is designed to be the most effective method of instruction. It will include a teacher station, three group stations, and 15 student stations. During a two-week unit, instruction may proceed as follows: The students will take a pretest on, say, Friday. They will be presented with a content overview by the teacher with the teacher station on

Monday. Tuesday, half of the students will learn at the student stations while the other students learn at the group stations. On Wednesday, the whole class performs a "wet" or hands-on laboratory experiment. On Thursday, the reverse of Tuesday occurs. This process continues until the final day of the unit during which the teacher will perform a wrap-up and the students would take a post-test. Besides the scanner, the individual stations will be networked to the teacher station. This will allow student mastery of objectives to be tracked, eventually (in 1989) being sent to the state level.

Following are some of the other functions available at the student station. Through the use of the function keys, the student will be able to:

1. access a pictoral/verbal glossary of terms and concepts
2. invoke a word processor
3. call up assignments and work on them
4. send questions or comments to the teacher station (the teacher may review these questions during the wrap-up or at some other opportunity).
5. print the screen at the teacher's laser printer (shipped via the networking interface.
6. pause the system (also called the bathroom function)
7. review previous discussion
8. inspect the objective mastery status. The student is told by the teacher of the requirements for mastery of objectives.
9. take notes.

A complete classroom management system to handle all three classroom configurations will be developed. Additionally, the necessary specialty programs and networking interface software will be developed. TLTG is still exploring the possibility of using the videodisc player in a distributed architecture, but will base the decision upon actual student alpha tests to determine duty cycle needs.

PROTOTYPE DEVELOPMENT

Three subcontractors developed a prototype of the system. The prototype hardware and courseware was completed on June 10 and demonstrated to members of the Texas legislature on June 11 at the Governors Office Science and Technology Conference at the Infomart in Dallas. The subject of the demonstration was the electromagnetic spectrum, which is half of the unit on light. Of interest here is the fact that the Texas Education Agency has granted to TLTG a reduction of the normally required 40% wet lab requirement to 20% in recognition of the fact that this medium has the capability of remarkable simulation of physical and chemical phenomena. TLTG learned through the prototype that the short time schedule and tight budget negatively impacted the quality of the instruction. The designers did not have enough time (six days) to produce a creative script. Some of the creative thoughts that did get produced during the scripting phase could not be implemented due to the fact that they called for computer-generated graphics and animation which required hours of time on expensive equipment. Future work, as TLTG decided should involve a partnership with a company which can effectively donate time on such expensive equipment.

DEVELOPMENT OF THE PILOT

The prototype was hard coded in C language. The unique architecture of the prototype system required special attention either through hard coding or through the hardware definition section of an authoring system. It was realized during the coding of the prototype, however, that the visions in the head of the designer got buffered by the envisioning ability of the programmers. The pilot will be developed using an authoring system so that the instructional designers' creativity can be more effectively utilized. TLTG made the decision to go with an authoring system which supports digital audio, level III videodisc, touch screen (regardless if touch screen is actually used in the pilot), exits to other MS-DOS programs, computer graphic/text overlay, and robust instructional design. TLTG would eventually like to take advantage of CD-ROM or CD-I technology, speech recognition, and expert systems both in content support and in pedagogical inferencing (Brown, Miller, and Swigger, 1985).

The development of the pilot will involve orchestration of a diverse spectrum of talent. Due to the non-profit status of TLTG, much of the work can be performed at reduced cost. Following are several of the tasks involved with the pilot development:

1. Archive video collection. At rates averaging $250 per foot from many commercial science archive sources, costs can run prohibitively high. TLTG has located about 20 sources (and will locate more) of free-release video. Sources include the American Chemical Society, the American Physical Society, NASA Free Film Library, Texas school districts, Texas Community Colleges, military video libraries, military contractor libraries, and public broadcasting stations.
2. Alpha testing in the participating districts. All 11 participating districts will have at least one interactive videodisc system located in the classroom of a teacher which will be piloting the course. As the first press of a unit is completed, the teacher will, during her free time, test the unit with students and report observations and data back to TLTG. Vendors pay large amounts of money to set up test sites for technology-based instruction. Since the schools are very much a part of this effort, they are absorbing the cost of this testing.
3. Development of a teacher manual. Not only is this manual going to assist the teacher with the technology based instruction, it will also be used as a curriculum planner, content reference, laboratory exercise guide, and homework source. The development of this and the other manuals can run upwards of a million dollars. TLTG expects to work with a national publisher (with an investment in technology) in order to develop the printed materials. This should be a mutually beneficial arrangement.
4. Development of a student manual. The student will have a manual which contains reference a reference note section, a workbook section, and laboratory exercises. If the courseware and the manual prove to have stand-alone capability, the state of Texas could adopt the course and distribute it to the other 1000 school districts in the state. Eventually, the state could even fund additional projects instead of purchasing textbooks. Due to the multi-year adoption cycle of textbooks in the state, this should prove beneficial, since course updates would be much more inexpensive than the re-adoption and distribution process currently in place.
5. Development of a full set of laboratory experiments which parallel the course and take advantage of existing school district equipment and supplies. Due to the reduction in the state 40% laboratory requirement in favor of the simulation, this should reduce laboratory equipment costs for the districts.
6. Teacher training. The teachers involved with the pilot (approximately 50) will be attending periodic workshops during the development in order to take part in developmental discussions and to receive instruction so as to be comfortable with the system before the pilot. The teachers can receive either Advanced Academic Training (AAT) credit or college credit for the training program.
7. Preparation for the independent evaluation. TLTG is using the Educational Productivity Council EPC at the University of Texas at Austin to conduct an independent evaluation of the pilot. Student learning and motivation, teacher adaptation, and dozens of other data types will be measured, compared with control classes, and cross-correlations performed to allow school district to determine which configuration of class is best for them (results will probably differ depending on the particular school learning environment). The EPC has a program which required six man-years to develop which will perform this analysis (with slight modification). TLTG will work with the EPC to insure that the classroom management system's data is in a format which can be shipped via modem to from the district to Austin for the evaluation.
8. Assisting the districts with the facilities alterations and

equipment configuration. The typical physical science laboratory has faucets wither located in perimeter walls or in about six tables in the middle of the classroom. The latter configuration requires the district to equip a separate classroom (except for the teacher station only) for the computer equipment. TLTG will also assist the districts with determining the most appropriate lay-out for the classroom.

As of July, 1986, plans called for a pilot of the semester of chemistry to occur during the Fall of 1987 and the semester of physics to occur during the spring of 1988. In the interest of devoting an appropriate amount of time to development of the courseware, both pilots may be conducted during the spring of 1988. The eleven participating school districts will be piloting anywhere from two teacher stations to two full classrooms.

CONCLUSION

It has become apparent that TLTG, being a non-profit organization and acting as an interface between industry and the school districts, is in a unique position to assist in the development of technology based curricula. The school districts participated in the design and development, actually telling the vendors what they wanted. Instead of the school being the victim of a vendor, as many indicated, the districts have been participating in a joint venture between private industry and education. More of these projects will occur. NSCCEF is establishing a northwest node, as the second in a near-term goal of five seed projects in the nation. Each node will have open communications, hopefully even sharing courseware (which will be compatible). This may just be a start, as viewed from an observer in the year 2016. Perhaps in thirty years, we will have not only developed complete courseware for K-16, but will have linked together these islands of knowledge in such a way that a student learns the information when needed during a motivating period of instruction.

BIBLIOGRAPHY

1. Brown, Stephen F., Miller, James R., and Swigger, Kathleen M., "Intelligent Tutoring of Physics Via Personal Computer--Emphasis on Learning", Proceedings of the 26th International Conference of the Association for the Development of Computer-Based Instructional Systems, Philadelphia, March 25-28, 1985.

2. National Center for Education Statistics, The Condition of Education, 1983 Edition, U.S. Department of Education, 1983, p.188.

3. National Science Teachers Association Survey, December 1982, reported in Hope Aldrich, "Teacher Shortage: Likely to Get Worse Before It Gets Better", Education Week, July 27, 1983.

Contact: Stephen F. Brown
Texas Learning Technology Group
P.O. Box 400
406 East 11th Street
Austin, TX 78767-0400

ADCIS Proceedings, November 1986

OUR VIDEOTEX IS O.K., HOW'S YOURS?

Elisabeth M. Craig - University of Tennessee at Chattanooga

Phil Gross - University of Tennessee at Chattanooga

ABSTRACT

The continuing progress made in developing software for the University of Tennessee at Chattanooga's Videotex Project is discussed in this paper. The programming modules which link the host HP3000 computer system to the AT&T FRame Creation system (Videotex device) are described. Other factors, such as the disparity between the original frames and the images on the monitor, are examined.

INTRODUCTION

The AT&T Frame Creation System, Series 500 (or FCS 500), a Videotex device, was given to the University of Tennessee at Chattanooga as a grant in January 1985. The system, which creates high resolution graphics, was discussed in the article "A Videotex in Your Future?", presented at the ADCIS conference in New Orleans. A year ago we blithely wrote that by the end of Summer '85 we intended to connect our FCS 500 system to the host Hewlett-Packard 3000, Series 48 computer. This article explores reasons why this process took longer than anticipated.

For those not familiar with the term Videotex, it is used to describe a type of computer which is capable of creating and transmitting high resolution graphics in the form of frames or screens. Many airports and hotels use monitors which are linked to this type of computer to display information, pictures and graphic reproductions.

Videotex uses the North American Presentation Level Procotol Syntax (NAPLPS), pronounced as "nap-lips", which was developed by the American National Standards Institute (ANSI) and the Canadian Standards Association (CSA) in the early 1980s. In a nutshell, NAPLPS is a byte-oriented language which provides a very efficient means of coding graphic displays. Also, it permits user input from a terminal device. Thus, a shopper in a mall can indicate that he/she wishes to see the map for a specific area of the mall; through NAPLPS the Videotex device is able to supply the requested display (or frame).

These frames, once created, are stored in files on either 8 inch floppy diskettes or on a host disk drive. These files may be used with any terminal display device that understands the NAPLPS protocol. Files can be called, for display, either from the FCS 500 itself, or from a host computer. (When talking about connecting the FCS 500 to a "host" the tendency is to think in terms of a main-frame system. However, several institutions have successfully used a micro as a host; this adds the advantage of transportability.)

In addition to the FCS 500, we received several AT&T Spectre terminals, which are designed specifically to display frames created on the FCS 500 and transfer them to a monitor. (Spectre terminals are not equipped with monitors).

After receiving the equipment, we spent some time in learning how to use the system and quickly became intrigued by the high resolution graphics that it allowed us to design. Despite the high interest, we found ourselves unable to break through the barriers which prevented us from linking the FCS system to a host computer, in this case a Hewlett-Packard 3000, Series 48.

Since the FCS 500 basically was reduced to a stand-alone device without this link, it was vital that we solve the problem. Lacking this link, we were unable to display the frames on monitors. Because the Videotex system was too delicate and too heavy to move, we were restricted to one viewing location, namely the room where the FCS 500 was situated. This was particularly distressing since our long-range plans called for us to have monitors at strategic locations on campus so that students could receive information about closed classes, campus activities and the weather.

Also, with host capability, it would be possible to create a data base to receive answers entered by the students. For example, an instructor could design a series of frames, each displaying an exam question (multiple choice or otherwise). The student would enter his/her answer; the software and Spectre terminal would take the student´s response and send it to the host for comparison with right answer in the data base.

SOFTWARE

In October 1985, Phil Gross returned to UTC to work as a Systems Programmer. In January he attended a conference in Chapel Hill, North Carolina, which brought together recipients of the FCS equipment. Based on his prior experience and the knowledge gained from this conference, he finally made the vital connection to our Hewlett-Packard system. Since then he has written program modules to both store the frames on the host and to transfer these frames to a display monitor via a Sceptre terminal. Now, he is working on modules which will allow users to select items from a menu in order to display frames stored on the HP3000.

To assist the programmer, AT&T developed the Frame Creation Interface Protocol (FCIP), which is compatible with either a 7-bit or 8-bit environment. It is expressly designed to facilitate the transfer of NAPLPS items between the host and the FCS 500. While FCIP has many useful features, it is not designed to support all the particular requirements of specific host computers. As will be discussed in the section ´Important Considerations´ some problems, therefore, were experienced in using this protocol.

At UTC, software has been written to make the frames transferable from host to FCS 500 and vice-versa via the FCIP protocol. The programs/modules are written in the System Programming Language (SPL), which is HP 3000-specific. The UTC programs are available to those having a host HP 3000 system. Because of the size of the programs and because they are written in SPL, a program listing is not included in this paper. Instead, a schematic of the three software modules which are being developed, is shown in the diagram on the next page.

Module 1, which accomplishes the transfer from FCS 500 to the host, provides the following capabilities:

1. entering and exiting the FCS 500 graphics editor;
2. displaying NAPLPS coded files that are stored on either the host system or FCS 500 diskettes;
3. listing Videotex file names on both computers;
4. transferring files in both directions between the two machines;
5. deleting files on both computers;
6. showing FCS 500 system status and information displays.

Module 1 has been completed, and the remaining two modules are expected to be finished by June 1986.

IMPORTANT CONSIDERATIONS

The Frame Creation Interface Protocol (FCIP) of the FCS 500 is not very friendly. When sending a response to the host (HP 3000), it does not attach a carriage return or any other character to the end of its data. This means that a read to the terminal will cause the program to "hang" when in FCIP mode, that is, the HP 3000 is in an I/O suspend state.

Fortunately, the FCS 500 does have one characteristic that allows the problem to be side-stepped: when it receives a command that it does not

Diagram Showing Software Modules Needed to Transfer Frames from Host to Monitor

Numbers shown in parentheses refer to modules
in Software Layout

Physical Layout

```
                   File transfer
 ____________                         ______________________
|            |                       |         (2)          |
| FCS 500    |  <------------------> | Host  HP3000         |
|____________|         (1)           |______________________|
                                      (3) /          \ (3)
                                |----------/-|    |-\----------|
                                |   Spectre  |    |  Spectre   |
                                |____________|    |____________|
```

Software Layout:

```
   |------------|         Module 1:
   |            |         Transfers frames from FCS 500
   |  Module 1  |         to host (HP 3000)
   |____________|

   |------------|         Module 2:
   |            |         Maintains "sequence files"
   |  Module 2  |         that specify the display order
   |____________|         of frames.
         |
         |
   |-----|------|         Module 3:
   |            |         Controls Spectre terminals,
   |  Module 3  |         using sequences from Module 2
   |____________|         to display frames on a
                          monitor and process user input.
```

---

understand, it responds with an "unprocessed" (U) transaction. The transaction contains the misunderstood command itself. We have developed the following procedure to get around the problem. When invoking FCIP mode, Module 1 instructs the terminal driver to go into "unedited" mode and then defines decimal 23 (ETB) as the record termination character (normally a carriage return). This is necessary for two reasons:

a) it allows all characters to be transmitted from the FCS 500 with no action being taken on them by the terminal driver (backspace, cancel, etc.).

b) it allows the ETB character to terminate input rather than the carriage return. NAPLPS coded files can contain the carriage return character as part of their data. The ANSI standard (X3, 110-1983) covering the Presentation Level Protocol Syntax (PLPS) states that the ETB character is among characters reserved for such usage.

Every time a FCIP transaction is sent to the FCS 500, the ETB character also must be transmitted at the end. The FCS 500 does not understand it, and, therefore, echoes it back in a "U" transaction after the desired data. When the driver sees this character, it terminates the read. Clumsy, but it works.

USING MONITORS

Flip Wilson's saying "what you sees, is what you gets" does not necessarily apply to frames being displayed on a monitor. There are often substantial differences between the frame on the AT&T equipment and its image on the monitor. At the ADCIS conference in New Orleans, a seminar on using computers and video equipment to enhance lectures and discussions was presented. Lynn

Churchill, Stephen Graham and Harold Modell moderated the sessions. For those thinking of developing software or frames to be displayed on monitors, our experiences and those of this expert panel indicate there are some pitfalls:

1. Because of luminosity, colors may blend toward black, so items that appear as dark on a CRT screen may be displayed as black on the monitor.
2. Color monitors project a dimmer image than the original, so all colors are muted on the monitor.
3. Since about 10% of the population is color blind red and green should be used with discretion.
4. Text should be restricted to 40 characters per line, because many monitors cannot handle 80 characters.
5. Because of their high legibility, colors such as amber, white, green and cyan are the best choices for text.
6. Text may blur, show color bleeding or be distorted when shown on a monitor.
7. Because of technical differences between the host system and monitor equipment, software may be system dependent and may not transferable to a monitor.
8. If working with FCS 500, it is unwise to exceed 6,000 bytes per frame as the FCS system may not keep a frame greater than this size.

In other words, do not assume that frames which look good on your CRT screen, will display equally well on a monitor - test them. Other good tips include designing or buying a hand-held switch so you can shut the monitor off when talking. Also, carry spare adapters and connectors in case the ones in use break down.

BLINKING FRAMES

One of the options available on the FCS 500 is to create frames containing blinking images. While the method for making one object blink is simple, synchronized or sequenced blinking of several objects is complicated, and requires careful planning to achieve the right effect. At a March 1986 FSC 500 Users' Conference, Mike Sutherlin (AT&T) described the FCS 500 technique to create a blinking screen of an arrow composed of three rectangles with a triangle at the end. To do this, it is necessary to construct four frames, one for each of the four geometric shapes. Blinking frames should be used with discretion as they can rapidly lose their uniqueness while research indicates the flashing lights may induce epileptic seizures.

AT&T FRAMES

Some of the best FCS 500 frames are those developed by Mark Zweigler (AT&T). Mark has been working with Videotex systems for seven years and combines a knowledge of this equipment with a high degree of graphic artistry. The frames, copyrighted by AT&T, are outstanding for their visual effects. He has developed a method which provides an "air brush" effect by constructing cylinders composed of slender rectangles of white, black and shades of grey. When these rectangles are put side-by-side to form the cylinders, a three-dimensional effect is achieved.

SUMMARY

Working with a Videotex system requires a wide range of talents, as evidenced by the topics covered in this paper. The best approach is a group one in which experts of networks and computer architecture join forces with programmers and graphic artists. However, there are ways for the individual to obtain help and encouragement. In March 1986 a conference of recipients of FCS 500 was held at the University of West Virginia. This meeting allowed an exchange of information and software which encouraged those attending to find new ways to employ Videotex in a campus environment. It is hoped to establish a formal users group and to publish a newsletter.

Although equipment such as Videotex provides a challenge, the resulting graphics are well worth the effort. There also is the satisfaction gained from acquiring "hands-on" knowledge of a new technology. We hope that our experiences will be helpful to other institutions who are using FCS 500 or other Videotex systems. In the next few months we will be expanding our knowledge and would be happy to share it.

REFERENCES

Churchill, L., Graham, S. and Modell, H. Using a Computer and Video to Enhance Lecture/Discussions. Seminar presented at 27th ADCIS Conference, New Orleans Louisiana, February 3-6, 1986.

Craig, E.M. (1986). A Videotex in Your Future? 27th ADCIS Conference Proceedings, New Orleans, Louisiana February 3-6, 1986.

Sutherlin, M., District Manager, AT&T 60 Columbia Turnpike, Morristown, N.J.

Zweigler, M. AT&T Information Systems, 60 Columbia Turnpike, Morristown, N.J.

Contact: Beth Craig
410 Hunter
University of Tennessee at Chatanooga
Chattanooga, TN 37401
615-755-4389

The Effectiveness of Interactive Video in Teaching Information and Applications

David W. Dalton

Indiana University

ABSTRACT

Although recent research has shown that interactive video is an effective system for the delivery of instruction in a wide variety of information-intensive, visually-oriented learning contents, little research has been devoted to ascertaining the effectiveness of more application-oriented interactive video programs. This study compared the effectiveness of an application-intensive interactive video treatment with an information-intensive interactive video treatment and a control on both recall of information and application of that information, as well as attitudes toward instruction. The 116 subjects were randomly assigned to their respective treatment groups where they completed a lesson on basic economic theory. Results from the study indicate that the application-intensive treatment was as effective as the information-intensive group in teaching the information presented. However, the application-intensive group was significantly more effective in teaching application skills. No significant attitude differences were found among treatment groups.

INTRODUCTION

Because of its ability to merge the interactivity and answer judging capabilities of the microcomputer with the realism and action afforded by video images, computer-assisted video instruction (CAVI) is often heralded as the ultimate instructional delivery tool (Thordildson, 1983). In addition, there is evidence to suggest that the variety of learning stimuli present in CAVI lessons may enhance both short- and long-term recall of instructional material (Clark, 1984).

One of the main alledged adavantages of interactive video over conventional video as a delivery system lies in its ability to collect and judge learner responses and define an appropriate instructional path for each learner based on those responses (Nugent & Stone, 1980).

The use of this learning technology seems well-suited to learning tasks which involve visually oriented information, such as CPR training (Hon, 1982) or when the consequences of failure to learn are high, as in flight training. In general, the use of CAVI as an instructional delivery systems with information intensive, visually rich training has been well established (see, for example, Pawley, 1983).

However, there are many instructional setting, principally the public schools, where less focused, general educational programs have been successfully enhanced through the use of traditional instructional video. This type of video instruction has been used to provide a richness not avaliable within the context of the conventional classroom. In short, rather than delivering information, this type of video is often used to "set the scene" for teacher-led instruction that will later demonstrate some applications of the learning tasks in settings removed from the classroom. Although the value of video images in demonstrating these applications to supplement conventional teacher-led instruction seems somewhat obvious, there are unresolved questions as to the utility of this approach in interactive video instruction.

The present study compared the effects

of an interactive video treatment used primarily to present factual information with a CAVI program showing applications of that information on the learners' achievement of both recall and application questions, as well as attitudes towards instruction.

METHODS

Subjects

The subjects chosen for partcipation in this study were 116 ninth grade learners selected from six sections of an introductory economics class. The sample consisted of approximately equal numbers of males and females and was composed primarily of Anglo students, with some minority learners.

Materials

The basic instruction for this study consisted of a 30 minute video lesson on the concept of opportunity costs and the opportunity costs of buying on credit. The basic lesson was modified as follows.

Information-intensive CAVI lesson. In this treatment, the learners were shown scenes of the video relating to specific factual information, primarily definitions of lesson concepts. After each video segment, the learner was asked a recall question on the information just presented. For example, after a video segment that defined and explained the concept of the opportunity costs of buying on credit, the computer asked the learner to input a missing key word from the definition previously given in the video. This treatment contained a total of ten such segments.

Application-intensive CAVI lesson. In this treatment, the learners were shown applications of the definitions of the concepts. Again, after each video segment, the computer asked the learners to input a response. In this case, learners were asked to respond to open-ended questions such as, "If you decided to save your money to buy a car, what would be the opportunity costs of that decision?" This treatment also consisted of ten instructional segments.

Control. In this treatment, learners were shown the entire video in a linear fashion with no questions from the computer.

Dependent Measures

After their respective treatments, all learners were given a print-based posttest that contained two parts: a recall posttest and an applications posttest. The recall posttest consisted of ten fill-in the blank items. Split-half reliability of this instrument was .77 using data obtained during the study. The applications posttest also contained ten fill-in the blank items and had a split-half reliability of .72, again using data obtained during the study. The reliability of the combined posttest was .84.

In addition to the achievement measures described above, the learners were also given an annonymous print-based twenty-item Likert-type survey designed to assess their attitudes towards the instruction. The split-half reliability coefficient of this instrument was found to be .94 using data obtained during the study.

Procedures

Prior to the beginning of the study, learners were designated as relatively high or low in prior achievement based on a review of ninth grade CTBS social studies scores. The learners were then randomly assigned to their respective treatments where they completed the instruction. Immediately following the lesson, the learners were given the posttests previously described.

Data Analysis

The design of this study was a completely crossed 3 x 2 x 2 treatment-by-sex-by-ability factorial design, featuring three levels of treatment, information-intensive CAVI, application-intensive CAVI, and control, two levels of sex, male and female, and two levels of ability, relatively high and relatively low.

Achievement and attitude data were analyzed through fixed-effects ANOVA procedures.

RESULTS

The results for the recall posttest were as follows. The mean of the application-intensive treatment group was 80.33%, while the mean of the information intensive group was 77.73% and the mean of the control group was 69.13%. The means of the application intensive group and the infromation intensive group were significantly different than the control ($.001 < p < .01$). However, the means of the application intensive group and the means of the information intensive group were not significantly different ($p > .05$).

The mean of the female learners was 80.06%, while the mean for the male learners was 70.47%. These means were significantly (.001 < p < .01). The mean of the relatively high learners was 80.06% and the mean for the relatively low learners was 62.73%. These means were also significantly different (.001 < p < .01).

The results for the application posttest were as follows. The mean for the application intensive treatment group was 67.00%, which differed significantly (.01 < p < .05) from the information-intensive and control groups, both of which averaged 62.47%. Male and female learners both averaged 64.00%. The mean of the high learners was 64.47%, while the mean of the relatively low learners was 62.47%. These means did not differ significantly (p > .05). There was also a significant treatment-by-achievement interaction on the application posttest, with low learners in the information-intensive group scoring significantly lower (p < .001) than their counterparts in either the application or control groups.

The mean for the application intensive, information intensive and control groups were 62.57%, 64.63%, and 64.66%, respectively on the attitude survey. These means were not statistically differnt (p < .05). The means for male and female learners were 63.22% and 64.45% respectively. These means were also not statistically differen (p > .05). The means of the high and low learners were 63.20% and 66.33% respectively. These means were not significantly different (p > .05).

DISCUSSION

There are three important results from this experiment that warrant discussion. First, the means of both interactive video treatments were significantly greater than the mean of the video control group. This result supports the research which suggests that more interactive forms of video instruction produce higher levels of comprehension (Saloman, 1983).

Specifically, learners in the interactive video treatments were called upon to answer questions on the viewing content as the lesson proceeded. These questions apparently caused the learners to attend to their lesson more intently than the stand-alone video lesson without these questions. This increase in attention and the corresponding increase in mental effort was likely the cause of the superior performance of learners in these treatments.

Second, learners in the application intensive interactive video treatment outscored learners in the information intensive group on the applications posttest. This result suggests that focusing on application-level content is effective in teaching these application-level skills. This result is not surprising in terms of the objective of this lesson version. Here, learners were provided with direct instruction and shown examples on the application s of the lesson concepts. This focus on application was not present in either of the other treatments.

This result, coupled with the fact that both CAVI treatments were equally effective in teaching recall skills, the use of application-intensive CAVI, if properly structured, seems to be an effective way to teach both levels of skills simultaneously. The implications of this conclusions should be of particular interest to a wide variety of current and potential CAVI lesson designers.

To date, much of the research and development activities associated with CAVI have concerned information-intensive lessons, such as that described in this study. However, because of the technical capabilities of the interactive video medium, many authors have actively advocated a shift in focus from this type of lesson to lessons that foster higher-level learning outcomes through the use of case studies, simulations, and similar instructional strategies. The results of this study strongly supports this arguement.

Finally, no significant differences were observed on the attitude survey employed in this study, suggesting that all three interventions were equally well received by the sample of learners involved. Although not extremely unusual, all three treatments were somewhat differenet than the normal classroom routine experienced by these learners. Therefore, all learners may have experienced an equal improvement in attitude. However, it is important to note that the time frame of this study was relatively short. Therefore, it seems plausible that the instructional intervention employed was simply not strong enough to promote a noticeable improvement in learner attitude toward instruction.

In summary, this study lends support to the notion that interactive video can be used to provide effective instruction on both information- and application-intensive content. In addition, this study lends support to the notion that a focus on higher-level learning experiences, such as simulations and case studies, need not come at the expense of basic information-level skills.

Future research should explore ways in which interactive video might be designed to teach other types of learning skills, including concepts and cognitive strategies through the use of simulations and "scene setting" presentations such as the one employed in this study.

REFERENCES

Clark, J. (1984). How do interactive videodiscs rate against other media? Instructional Innovator, 29(6), 12-16.

Hon, D. (1982). Interactive training in cardiopulmonary resusitation. Byte, 7(6), 108-120, 130-138.

Nugent, G. & Stone, C. (1980) Videodisc instructional design. Educational Technology, 20(5), 29-32.

Pawley, R. (1983). It´s becoming an interactive world. Educational and Industrial Television, 15(6), 80-81.

Saloman, G. (1983). Television watching and mental effort: A social psychological view. In J. Bryant & D. Anderson (eds.), Children´s understanding of television. New York: Academic Press.

Thordilkdson, R. (1983). A microcomputer/videodisc package to teach the retarded. Educational and Industrial Television, 15(6), 40-42.

ADCIS Proceedings, November 1986

QUALITY CONTROL IN THE DEVELOPMENT OF COURSEWARE UTILIZING COMPUTER-BASED INTERACTIVE VIDEO

Richard A. Dassance
Center for Instructional Development and Evaluation
The University of Maryland University College

Abstract

Formative evaluation procedures used when developing courses employing computer-based interactive video (CBIV) must be more extensive than the procedures for evaluating typical computer-based instruction. The formative evaluator has to plan both for the inability to revise a mastered videodisc and for the numerous review categories necessary for an audio/visual/textual/individualized production. Eight categories were determined necessary for the formative review of CBIV courseware. These are Content, Instructional Approach, Interaction, Aesthetics, Nontextual Thinking, Language, Screen Display Rules, and System Usage. Some or all of these categories would be the focus of review, revision, and review again at the following five time periods: creating the initial design, writing treatment plans, developing storyboards, producing any audio or visual material, and programming the lesson. After this quality-control procedure, CBIV courses are then ready for pilot and field studies.

The Problem

With the advent of interactive videodiscs, the course development process has more people doing a greater variety of tasks than development of traditional instructional materials has required in the past. The need for establishing, coordinating, and controlling the quality-seeking process has increased.

Typically, formative evaluation has been one of checking and revising an almost-finished product to determine how well it works with learners representing the target population (Dick & Carey, 1978). Unfortunately, computer-based interactive video (CBIV) does not offer the luxury of inexpensively revising mastered discs and programmed graphics. One-to-one formative evaluations and field studies are still valuable, but often come too late if they are to be used as a first step in the quality-assessment process.

Stufflebeam (1974) has expanded the definition of formative evaluation to be applicable to each step of the design and development process. He thinks that formative evaluation "serve(s) the needs of decision making about program development" (Gagne and Briggs, 1979, p. 289). That definition appealed to our course development team. We wanted a formative evaluation procedure to assist the decision making process and to assure quality from initial design, through the creation of storyboards, and into production and programming.

To that end, we looked for a quality-control procedure that was comprehensive, systematic, usable, and useful. Comprehensive, to cover all categories of CBIV concerns that usually are not issues with most computer-assisted instruction (CAI). Systematic, because so much material would be reviewed, revised, and again reviewed, and because so many "sets of eyes" would be needed. Usable, in the sense of not being overwhelmingly frustrating and burdensome to the development team. And useful, by being a cost-effective way to make a better product.

The suggestions for quality control were mostly evaluation criteria to be used in summative and traditionally formative evaluations. Nowhere in our searching did we find a systematic procedure that would guide us to establishing a comprehensive set of standards applicable to CBIV and meeting our needs. Therefore, we decided to develop our own quality-control procedure.

One Solution

In creating a quality-control procedure, we wanted to insure a systematic approach to all the courseware concerns that are relevant to computer-based interactive video. We first established general categories for review as a way of generating the standards. The primary reviewers for each of these categories became advocates for establishing standards well in advance of the creation of treatment plans, storyboards, and the programmed lessons.

As standards were developed, they were included in our Project Guidelines Notebook. Standards were written lightly in pencil (rather than engraved in stone), allowing later re-evaluation and renegotiation. Many of the team members had a hand in developing the guidelines initially; all had the responsibility of helping to suggest additions and revisions.

This paper will show which categories we included in our guidelines notebook, when review and revision were needed for each of those categories, and who was most appropriate as the expert reviewer and advocate for quality control for each of those categories.

Review Categories

Content. The primary task for this category was guaranteeing content accuracy. However, the content reviewers were asked to do other tasks also. One such task was creating and reviewing the standards for the language to be used in the course. For example, we wanted a consistent use of terms and concepts throughout all the lessons.

Another task for the content reviewers was to insure a logical flow in the material. That is difficult in a non-linear, modularized course. Non-linear courses also require assurance of content independence of the different modules which explicitly do not require any prerequisite knowledge found in other modules.

Standards were needed for depth of coverage. Objectives needed to be taught, yet we did not want to go beyond those objectives and the scope of the course. To maintain a consistent depth of coverage through all the modules, we developed guidelines.

Instructional Approach. Guidelines for the instructional strategies and for the course structure needed to be established and followed. The course structure would include length of the lessons, number of learning points to be covered in each, what metaphor would carry the content, and what metaphor would be used for system usage. (A system-usage metaphor would be, for example, laying the course out like a book with chapters, a table of contents, a glossary, and an index.)

Another area of concern to the instructional-approach reviewers was the general flow of the course. They would be concerned about the story line (for consistency and appropriateness) and the transitions (such as the effectiveness of the advance organizers).

Interaction. An interaction course needs to be assured of having an appropriate interaction beyond simple "page turning." This is a category which is usually everybody's concern, so it is not explicitly reviewed. We asked our interaction reviewer to be responsible for the advocacy, establishment, and review of standards for these three areas:

- Feedback - non-threatening and audience appropriate.
- Branching and Returns - optional or mandatory; and alternative approaches (or not) to previously covered material.
- Practice and Testing - location of practice questions; normal validity and reliability questions.

Aesthetics. The reviewer for this category needed to follow some subjective internal guidelines that were not easily standardized. The first area for this person was visual aesthetics (e.g., Is each frame appropriately attractive, open, and balanced?). The second area was entertainment (i.e., determining the client's fine line between creating what they consider to be an enjoyable presentation and what they think is a frivolous waste of training time). The final area is the use of humor (How much? Tasteful? Regional? Dated? Appropriate sophistication level?). A wise and trusted person had to make these subjective decisions.

Non-textual Thinking. It is very hard for some people to convey ideas without using words. The reviewer for this category helped the scriptors think visually and auditorally. Visual thinking implies the use of charts, diagrams, still and motion pictures, animation and icons. Auditory thinking includes the use of background noises, sound effects, theme music, and voice tones. Having a reviewer with this concern in mind seemed to increase the use of non-textual media.

Language. A single reviewer edited the text for consistency in writing style, syntax, and wording formats for definitions, titles and labels. Standardizations needed to be developed for abbreviations, capitalizations, and special expressions that were used. A final concern for this reviewer was to check that modularized lessons were not dependent on other non-prerequisited lessons for explanations to acronyms and special terms.

Screen Display and Audio Rules. This reviewer's task differed from the aesthetic concerns in that for this category there were very clear, non-subjective rules and guidelines to be followed. Before storyboards were written, standards were explicitly defined for things such as which colors were to be used for what purposes; text fonts; locations of text, graphics, menus, and operations information; and timing as to when sound or animations were to begin after a frame appeared.

System Usage. This reviewer looked at the system from two perspectives: learner use and system framework. For system framework concerns, the reviewer checked log-on and log-off procedures, automatic routings, and access to lessons and HELP. In general, the reviewer was concerned with the course structure as created by the systems programmer. This basic framework had only to be checked out on line once.

The other perspective for this reviewer was learner use. This person evaluated how easy the system was to use by looking at such factors as consistency in the use of commands and maintenance of the frame of reference (i.e., the learners should always know where they are in the course and what they are supposed to be doing).

Of course, all movement through the course had to occur as designed. By using every branch, command, and option, most of the bugs were discovered.

Formative Evaluator

The formative evaluator was a team member with on-going course development responsibilities. He was not an external agent who was limited to objectively applying a list of standards after the product was developed and almost ready to be delivered to the client.

The primary task for the formative evaluator was to be an advocate for quality. This meant being the person who kept the development team aware and concerned about reviews and revisions. The formative evaluator helped to determine what should be reviewed, by which criteria, when and by whom. This task included performing periodic validity checks of the quality-control process.

The formative evaluator reviewed the administrative tasks related to quality control. This included looking at the project timeline to ensure built-in review-and-revision periods; to be an advocate for the check procedures for establishing, communicating, implementing, and re-evaluating the various standards and guidelines; and to help choose, schedule, and supervise the various reviewers.

Reviewing Assignments

There were five critical times when reviews seemed appropriate. These were at the end of the following five activities:

- Initial Design - content outline, goals and objectives, instructional method, and general course-development guidelines.
- Treatment Plans - detailed descriptions as to how each lesson is to be handled.
- Storyboards - contain everything the production people and programmers need.
- Production - producing the audio and visual material that is not computer generated.
- Programmed Lesson - everything is put on line and combined with the produced audio and visuals.

In reality, some categories were reviewed by staff members who were not always the most expert and sometimes had to cover two categories at once. Ideally, we tried to use these reviewers:

- CLIENT - person or people representing the contract funder. Hopefully, their reviewers are always the same people so as not to have constantly changing standards of acceptability by the funder.
- SME - Subject Matter Expert, determined specifics as to what was to be taught in the course.
- BACK-UP SME - another content expert who reviewed the SME's initial material, such as the content goals, objectives, and outline.
- CHIEF ID - "master" or most trusted instructional designer.
- ID - instructional designer, developed the particular lesson being reviewed.
- PROGRAMMER - chief programmer, responsible for the software system design for the project.
- QA - person regularly does "Quality Assurance" (or QA) for all CAI or CBIV projects; has an expert eye for catching bugs in the system's usage and violations of screen-display rules.
- ARTIST - person with a good eye for aesthetics and visual thinking.
- EDITOR - person good at syntax and grammar; same person edited all lessons.

Fortunately, people were not needed to fill all of the nine reviewing positions to check all eight categories at all of the five times. the matrix in Figure 1 shows when reviews were needed and by whom. (An X in a matrix cell means no

review needs to be done for that category in that time period.)

	INITIAL DESIGN	TREATM'T PLANS	STORY-BOARDS	PRODUCT'N	PROGRAM'D LESSON
CONTENT	- CLIENT - BACKUP SME	- CLIENT - SME	- SME	- SME	- SME
INSTRUC-TIONAL APPROACH	- CHIEF ID	- CHIEF ID	- CHIEF ID	X	- CHIEF ID - ID
INTER-ACTION	- PROG'R	- PROG'R - CHIEF ID *	- PROG'R - CHIEF ID *	X	- CHIEF ID - ID
AESTHETIC	X	X	- ARTIST	- ARTIST	- ARTIST
NON-TEXTUAL THINKING	X	- ARTIST	- ARTIST	- ARTIST	- ARTIST
LANGUAGE	X	X	- EDITOR	X	- EDITOR
SCREEN DISPLAY & AUDIO RULES	X	X	- QA	X	- QA
SYSTEM USAGE	X	X	- QA	X	- QA

* OR THE ID IF SCRIPTORS ARE USED

Figure 1. Quality control for CBIV course development: when (horizontal axis), what (vertical axis), and who (cells).

Rather than having nine different people serve as reviewers, we used what people we had to serve as different "sets of eyes." When one person reviewed more than one category, then that person would schedule a separate review for each category in order to maintain a certain perspective and focus throughout each review.

It was only after the completion of these review steps that we determined the course ready for normal pilot and field studies with "real" students serving as reviewers.

Observations

As with the other aspects of course development, evaluation has to struggle with deadlines, schedules, and bottlenecks as when one person gets overloaded with work and gets behind. Review and revision are added tasks that--while making the produce better--make the job harder.

Everyone seems to agree that evaluation is a critical element in any project, yet not everybody seems to do it. We attributed the incorporation of evaluation into our project to one factor in particular: the project director's attitude. Because of her support and insistence, a systematic and comprehensive review-and-revision procedure became an integral part of CBIV course development.

The compromises made in our project were determined by the time considerations and client's priorities. It appeared that they wanted the course to have accurate content, to flow without bugs, and to be consistent in format, language, and punctuation. Revisions for improving instructional quality were made only at our own insistence--and only if time allowed.

The standards which were to guide everybody into predetermined conventions were often ironically determined after the fact. A problem or inconsistency would arise mid-development or during review; decisions were made; and a new standard would be born. Creation of standards for formative evaluation was an evolutionary process and will continue to be as instructional technology continues to change.

References

Dick, W., & Carey, L. (1978). The system of instruction. Glenview, IL: Scott, Foresman & Co.

Gagne, R., & Briggs, L. (1979). Principles of instructional design (2nd ed.). New York: Holt, Rinehart & Winston.

Stufflebeam, D. L. (1974). Alternative approaches for educational evaluation: A self-study guide for educators. In W. J. Popham (Ed.), Evaluation in education. Berkeley, CA: McCutchan.

Contact: Rick Dassance
Center for Instructional
Development and Evaluation
University of Maryland University Coll.
University Blvd. at Adelphi Road
College Park, MD 20742

ADCIS Proceedings, November 1986

THE EFFECTS OF LOCUS OF INSTRUCTIONAL CONTROL AND PRACTICE ON LEARNING FROM INTERACTIVE VIDEO

Michael J. Hannafin and Maryanne E. Colamaio

The Pennsylvania State University

ABSTRACT

The purpose of this study was to examine the effects of locus of instructional control and practice on the learning of facts, procedures, and problem solving skills during interactive video. Subjects were volunteers from graduate and advanced level undergraduate college classes. The instructional content was a 30-minute videotape, "Project Lifesaver," which was designed to introduce cardiopulmonary resuscitation (CPR). Students were randomly assigned to one of three instructional control groups: 1) designer imposed; 2) learner selected; or 3) linear (control).

The results indicated 1) significant differences between practiced and non-practiced information; 2) significant differences on posttest scores among locus of instructional control groups; and 3) a significant interaction between practice and type of learning.

* * * * * * * * *

Interest in the instructional applications of interactive video has grown dramatically during the past decade (Cambre, 1984; Ebner, et al., 1981). Interactive forms of video instruction have been described at various levels ranging from essentially uninterrupted linear video through fully conditional lesson execution where lesson sequence is controlled through the input of information external to the video itself (Daynes & Butler, 1984). For purposes of this paper, interactive video is defined as ". . . any video program in which the sequence and selection of messages is determined by the user's response to the material" (Floyd, 1982, p. 2).

Whereas interest in the application of interactive video has grown dramatically, little empirical data has been published to either guide or support design decisions (Hannafin, et al., 1985). Some proponents suggest that interactive video is a unique medium unlike either the video and computer component technologies (DeBloois, 1982). If true, then virtually no foundation exists upon which to base interactive video design decisions. Others have proposed a greater relationship among the evolved empirical research in the design of computer-based, video, and traditional instruction, in the design of interaction video (Hannafin, 1986). At the very least, accrued knowledge represents a more logical framework for testing the tacit assumptions of interactive video than simple acceptance of alleged uniqueness.

Much of the research in the design of computer-based instruction has focused on the locus of instructional control decisions. Locus of instructional control can range from fully learner controlled through complete imposed lesson control. Proponents of learner control point to individualization, increased sense of responsibility for learning. and the potential for optimal learning efficiency as support for transferring control of lesson components and/or sequence to learners (Bunderson, 1974; Laurillard, 1984; Merrill, 1975; Merrill, Schneider & Fletcher, 1980).

Recently, however, research findings containdicated unaided learner control of lesson activities (Steinberg, 1977) in favor of either imposed adaptive lesson control or learner control with various forms of embedded coaching or advising to the learner (See, for example, Ross & Rakow, 1981; Tennyson, 1980, 1984;

Tennyson, Christensen & Park, 1984). Factors such as the nature of the learning task, the age of learners, and the desired learning outcomes of the instruction operate interactively during computer-based instruction (Hannafin, 1984).

The relationship between type of learning and type of learner interaction during computerized instruction has also been studied. Some have suggested that interactive video is superior for higher-level learning--especially in the health sciences field (De Chenne & Evans, 1982; Hon, 1982; Levensen, 1983; Lyons et al., 1982). Others have suggested that the type, level, and location of interaction activities affects the magnitude and level of learning (Hannafin, et al., 1985). Precisely which types of interaction promote different particular of learning, however, remains unclear.

The purpose of this study was to examine the effects of varied interactive video lesson control options and of en-route lesson interactions on the learning of facts, procedures, and problem-sovling skills.

METHODS

Subjects

Forty-eight subjects, consisting of graduate and advanced level undergraduate college students, participated in this study. None of the participants had prior training in cardiopulmonary resuscitation (CPR) procedures, which were the focus of the instructional content of the interactive video lesson.

Instructional Content

The instructional content used for the treatment phase of the study was the American Heart Association videotape, "Project Lifesaver." "Project Lifesaver" is a 30-minute videotape presentation which is designed to introduce CPR. The instructional content was selected due to the extensive use of interactive video in health sciences and the requirement for multiple types of learning required in the training of CPR (e.g., Bidwell, Collins-Nakai, Taylor & Jensen, 1985; Harless, 1985; Hon, 1982; Rosenblat & Gaponoff, 1984).

The lesson was divided into four segments roughly equal in information density and duration: 1) "General Overview of the Heart Functions,: 2) "Airway Problems and Techniques," 3) "Breathing Problems and Techniques," and 4) "Circulation Problems and Techniques." Each segment presented facts and procedural steps to be applied in problem-solving situations.

A total of 12 questions were embedded during the lesson. In order to provide en-route practice of lesson content, the questions were distributed equally among factual information, procedural steps, and problem-solving applications. One of each question type was embedded at the end of each of the four lesson segments.

Instructional Treatments

Three locus of instructional control versions of the lesson were developed: designer imposed, learner selected, and linear.

Designer Imposed. Students in this group followed a pre-determined path through the lesson. After answering each embedded question, students were given knowledge of results of their response. At this point, they proceeded to the next segment if the answer was correct, or were branched to review only the video segment that related to the question if incorrect. After the review, the question was presented a second time. Again, knowledge of results was provided, and the lesson continued if the answer was correct. If incorrect on the second attempt, the correct response was provided before the lesson continued.

Learner Selected. Students in this group controlled their path through the lesson. At each point for which a designer imposed decision was enforced in the imposed control group, the student was permitted to make an individual control decision. Unlike the imposed control group, students were permitted to choose the order of the video segments. Whereas students were advised as to the recommended lesson sequence, the actual selection of sequence choices was made individually. In addition, after answering the embedded questions and given knowledge of results, students were given instructional control options. If the answer was correct, the student was given the option to review the video segment that pertained to the question or to proceed to the next part of the lesson; if incorrect, the same choice was provided. If the student chose to review the video segment, the option to repeat the question a second time was provided. If the student chose not to repeat the question, the lesson proceeded to the next part. If the question was chosen a second time and answered correctly, knowledge of results was provided and the lesson proceeded. If answered incorrectly on the second try, the correct response was provided and the lesson proceeded to the next part.

Linear (Control). Students followed a linear path through the lesson. After answering each of the embedded questions, students were given knowledge of results and proceeded to the next segment of the lesson. No options for controlling the sequence of the lesson, and no imposed decision for remediation or question repetition, were provided.

CPR Posttest

The criterion test consisted of 24 randomly ordered multiple choice questions which tested for knowledge of the information and procedures, as well as problem-solving applications of the knowledge presented during the lesson. The 12 embedded postsegment questions, distributed equally among facts, procedural steps, and problem-solving applications, were repeated on the posttest. In addition, 12 new questions were divided equally among facts, procedures, and problem-solving. Knowledge of results was not provided after individual posttest questions. Feedback in the form of total number of correct responses was provided upon completion of the posttest. The reliability coefficient for the 24-question multiple choice test was .61.

Dependent Measures

The dependent measures were scores on each of the posttest subscales: Facts, Procedures, and Problem-Solving. Separate scores were computed for each type of learning for practiced and non-practiced items for each type of learning. Practiced items were those embedded during the lesson; non-practiced items were not presented initially, but relevant instruction was presented during the lesson.

Design and Data Analysis

The study employed a 3 x 3 x 2 factorial design. One between subjects factor, Locus of Instructional Control, included three levels (Linear (Control), Designer Imposed, and Learner Selected). There were two within subjects factors: Test Scale which consisted of three levels (Facts, Procedures, and Problem-Solving), and Practice which consisted of two levels (Practiced and Not-Practiced).

MANOVA procedures were used to analyze the CPR posttest socres. Newman-Keuls contrast procedures, where appropriate, were used to examine pairwise differences.

Procedure

The 48 participants were randomly assigned to one of the three instructional control groups. The participants were introduced in general terms to the nature of the lesson but were told of neither their particular group assignment nor how their treatment differed from others.

All participants viewed the "Project Lifesaver" interactive video lesson in accordance with treatment group assignments. Upon completion of the lesson, the CPR Posttest was administered immediately.

All treatments were administered individually in a room designated for the study. The instructional treatments were systematically rotated to ensure balance across treatments and to avoid contamination of individual treatments over time.

Results

Table 1 contains the means and standard deviations for types of learning by treatment combinations. The findings are summarized by effect.

Locus of Instructional Control

Significant differences were found among the three levels of instructional control, F (2,45) = 5.02, $p < .01$. Both the Designer Imposed ($\bar{x}$ = 20.7) and the Learner Selected ($\bar{x}$ = 20.5) groups performed better than the Linear group ($\bar{x}$ = 18.4). No significant differences were found between the Designer Imposed and Learner Selected groups.

Practice

Significant differences were found between scores on practiced and non-practiced items, F (1,45) = 44.65, $p < .0001$. Practiced item ($\bar{x}$ = 10.8) were learned at a significantly higher rate than non-practiced items ($\bar{x}$ = 9.0).

Type of Learning

Significant differences were also found among the three levels of learning, F (2,44) = 6.90, $p < .002$. Procedural learning ($\bar{x}$ = 6.9) was greater than problem solving learning ($\bar{x}$ = 6.3), but not significantly different from factual learning. No significantly differences were found between factual and problem-solving learning.

Type of Learning x Practice

There was a significant interaction between Type of Learning and Practice,

Table 1

Means and Standard Deviations for Scales of CPR Posttest by Treatment Combination

Locus of		Facts		Procedures		Problem Solving		Totals	
Inst. Control		Pr	NotPr	Pr	NotPr	Pr	NotPr	Pr	NotPr
Designer									
Imposed	$\bar{x}$	4.0	2.8	3.7	3.5	3.9	2.9	11.6	9.1
(N = 16)	SD	(0.0)	(0.6)	(0.6)	(0.7)	(0.3)	(0.9)	(0.9)	(2.2)
Learner									
Selected	$\bar{x}$	4.0	2.8	3.7	3.6	3.5	2.9	11.2	9.3
(N = 16)	SD	(0.0)	(0.7)	(0.6)	(0.6)	(0.6)	(0.7)	(1.2)	(2.0)
Linear									
(Control)	$\bar{x}$	3.6	2.5	3.1	3.2	2.9	2.9	9.6	8.6
(N = 16)	SD	(0.6)	(1.0)	(0.9)	(0.7)	(0.9)	(0.9)	(2.4)	(2.5)
Totals	$\bar{x}$	3.9	2.7	3.5	3.4	3.4	2.9	10.8	9.0
(N = 48)	SD	(0.4)	(0.8)	(0.7)	(0.7)	(0.7)	(0.9)	(2.1)	(2.4)

F (2,44) = 30.74, $p < .0001$. Scores on practiced items were higher than non-practiced items for each type of learning, but the effects were proportionately greatest for factual learning and least influential for procedural learning. No other interactions were statistically reliable.

Discussion

The purpose of this study was to examine the effects of locus of instructional control and practice on the learning of facts, procedures, and problem solving from interactive video. The results suggest that either of the interactive versions of instruction was superior to the linear design, embedded practice exerts the most powerful influence among the treatment variables studied in learning from interactive video, and that the influence of practice is greatest for factual and problem-solving learning.

The differences found for locus of instructional control were interesting. Both of the "interactive" video lessons were superior to the linear video lesson. This finding is consistent with findings for computer-based instruction which suggest that adaptive designs, with sequence options either imposed by the designer or learner control with coaching for making choices, yield successful learning (See, for example, Ross, 1984; Tennyson, 1984). Practice, as defined in this study, was important for the learning of facts and problem-solving skills, but relatively unimportant for the learning of procedures. This effect is consistent with the hypotheses of early television researchers. For procedural tasks, visual images are important aids in that they illustrate for the learner the succession of steps to be followed (Chu & Schramm, 1967). In effect, a form of vicarious rehearsal may occur during which appropriate visually oriented procedures can be modeled and consequences observed. Video images may provide sufficiently strong instructional stimuli to illustrate visual procedures, and to encourage appropriate mental rehearsal, that physical practice in the procedures themselves is unnecessary in some cases (See also Heestand, 1978, Allen, 1957; Fitts & Posner, 1967).

Problem-solving, on the other hand, seems more sensitive to the influence of formal practice during interactive video instruction (Hannafin, Phillips & Tripp, 1986). Though the required procedures may be fairly well known, the selection and application of such steps to novel situations during controlled instruction is uniquely important. These findings offer support for the growing interest in interactive video simulations (Floyd, 1982; Harless, 1985; Hon, 1983; Levensen, 1983) where the practice and application of problem solving skills is the principal learning focus.

The impact of interaction during video lessons--the opportunity to respond and receive feedback and appropriate remediation or clarification--also provides support for some of the promise of interactive video. The findings of this study are consistent with previous research in which progressive interaction improved learning from video in a more or less linear manner: The more interactive instruction, the greater the learning of factual information (Schaffer & Hannafin, 1986). The present findings replicate, extend, and qualify this conclusion. This study replicated the effects using verbal information derived from an entirely different content and tentatively extended, earlier conclusions to include problem-solving. The effects, however, were not evident for procedural learning, thereby qualifying the scope of the inference. Based on both earlier research and the present study, interactive video instruction appears useful and effective for certain types of learning, but is likely unnecessary or even ineffective for others (Hannafin, et al., 1985).

A predicted interaction between practice and locus of instructional control was not found. Practice was hypothesized as most important for the learner selected group since the inherent sequence provided through imposed lesson control assured the remediation of mislearned information, the repetition of questions missed initially, and the provision of correct answers when needed. In the learner selected treatment, the implementation of these features was controlled directly by individuals permitting students to select when (or if) options would be utilized. It was presumed that appropriate practice and feedback would be most important for this group since individuals could use information derived during practice to self-regulate, modify subsequent decisions, and otherwide alter the manner in which choices were made. An inspection of the "path" followed by students in the learner selected treatment, however, revealed that only one of the 16 participants followed a lesson completion path other than the one "advised" initially. In effect, the operational differences between the designer and learner control groups were virtually eliminated during the study. The perceptions of control by the learner selected group, however, were likely altered (Tennyson, Christensen & Park, 1984).

The findings of this study provide additional support for the effectiveness of interactive forms of video instruction. Based on the few research studies reported to date, there is reason to be encouraged with the performance of interactive video. The study of interactive video, however, remains in its infancy. Whether the context for study is to be extrapolated from related research or a new research context is to be proposed to examine the presumed unique capabilities of the technology, disciplined inquiry in the study of interactive video is imperative. A considerable need for further study remains before the promise of the technology can be validated.

REFERENCES

Allen, W. H. (1957). Research on film use: Student participation. Audio Visual Communication Review, 5(2), 423-450.

Bidwell, C. M., Collins-Nakai, R. L., Taylor, M. B., & Jensen, W. (1985). Multidisciplinary team production of computer-based simulation to teach pediatrics. Journal of Medical Education, 60(5), 397-403.

Bunderson, C. V. (1974). The design and production of learner controlled courseware for TICCIT system: A progress report. International Journal of Man-Machine Studies, 6, 479-491.

Cambre, M. A. (1984). Interactive video. Instructional Innovator, 29(6), 24-25.

Chu, G. C., & Schramm, W. (1967). Learning form television: What the research says. Washington: National Association of Educational Broadcaster.

DeChenne, J. A., & Evans, R. (1982). Simulating medical emergencies. Instructional Innovator, 27(1), 23.

Daynes, R., & Butler, B. (1984). The videodisc book: A guide and directory. New York: John Wiley & Sons.

DeBloois, M. L. (1982). Principles for designing interactive videodisc instructional materials. In M. L. DeBloois (Ed.), Videodisc/microcomputer courseware design. Englewood Cliffs, NJ: Educational Technology Publications.

Ebner, D. G., Mannnig, D. T. Brooks, F. R., Mahoney, J. V., Lippert, H. T., & Blason, P. M. (1984). Videodiscs can improve instructional efficiency. Instructional Innovator, 29(6), 26-28.

Fitts, P. M., & Posner, M. I. (1967). Human performance. Monterey, CA: Brookes Cole.

Floyd, S. M. (1982). Thinking interactively. In S. M. Floyd & B. Floyd (Ed.) Handbook of interactive video. White Plains, NY: Knowledge Industry Publications.

Hannafin, M. J. (1984). Guidelines for using locus of instructional control in the design of computer-assisted instruction. Journal of Instructional Development, 7(3), 6-10.

Hannafin, M. J. (1986). Examining the empirical assumptions of computer-based interactive video. Educational Communication and Technology Journal, 33, 235-247.

Hannafin, M. J., Garhart, C., Rieber, L., & Phillips. T. L. (1985). Keeping interactive video in perspective: Tentative guidelines and cautions in the design of interactive video. In E. Miller and M. Mosley (Eds.), Educational media and technology yearbook. Denver: Libraries Unlimited.

Hannafin, M. J., & Hughes, C. (1986). A framework for incorporating orienting activities in computer based interactive video. Instructional Science, in press.

Hannafin, M. J., Phillips, T. L., & Tripp, S. D. (1986). The effects of orienting activities, rehearsal, and processing time during interactive video. Journal of Computer-Based Instruction, in press.

Harless, W. G. (1985-March). A new technology - a new education. Presented at the annual meeting of the Association for the Development of Computer-Based Instructional Systems, Philadelphia.

Heestand, D. (1978). The use of inserted questions in videotape programs. International Journal of Instructional Media, 7(2), 149-157.

Hon, D. (1982). Interactive training in cardiopulmonary resuscitation. Byte, 7(6), 108-138.

Hon, D. (1983). The promise of interactive video. Performance & Instruction Journal, 22(9), 21-23.

Laurillard, D. M. (1984). Interactive video and the control of learning. Educational Technology, 24(6), 7-15.

Leverson, P. M. (1983). Interactive video: A new dimension in health education. Health Education, 14(6), 36-38.

Lyons, C. Krasnowski, J., Greenstein, A., Maloney, D., & Tatarczuk, J. (1982). Interactive computerized patient education. Heart & Lung, 11(4), 340-341.

Merrill, M. D. (1975). Learner control: Beyond aptitude-treatment interactions. Audio-Visual Communications Review, 23(2), 217-226.

Rosenblatt, A. A., & Gaponoff, M. (1984). The microcomputer as a vehicle for continuing medical education. Journal of Family Practice, 18, 629-632.

Ross, S. M., & Rakow, E. A. (1981). Learner control versus program control as adaptive strategies for selection of instructional support on math rules. Journal of Educational Psychology, 73(5), 745-753.

Ross, S. M. (1984). Matching the lesson to the student: Alternative adaptive designs for individualized learning systems. Journal of Computer-Based Instruction, 11, 42-48.

Schaffer, L. C., & Hannafin, M. J. (1986). The effects of progressively enriched interactivity on learning from interactive video. Educational Communication and Technology Journal, in press.

Steinberg, E. (1977). Review of student control in computer-assisted instruction. Journal of Computer-Based Instruction, 3, 84-90.

Tennyson, R. D. (1984). Artificial intelligence methods in computer-based instructional design: The Minnesota Adaptive Instructional System. Journal of Instructional Development, (3), 17-22.

Tennyson, R. D., Christensen, D. L., & Park, S. I. (1984). The Minnesota Adaptive Instructional System: An Intelligent CBI System. Journal of Computer-Based Instruction, 11, 2-13.

ADCIS Proceedings, November 1986

INTEGRATING MULTI-MEDIA CBI APPROACHES INTO SIMULATOR TRAINING PROGRAMS

Michael J. Olson, Harold E. Popplewell

General Electric Educational Services

ABSTRACT

This paper describes a unique, high technology training system currently under development for the U.S. Marine Corps and the U.S. Navy's Space Warfare Systems Command by the General Electric Company. The system includes a broad range of instructional media, including conventional instruction, Computer-Based Instruction (CBI), two-dimensional simulation, interactive graphics and videodisc, a high fidelity training device, dummy mission equipment, actual Prime Mission Equipment (PME), and managed on-the-job training. The integration of these media and the myriad types of personnel required to develop such a system is a complex challenge from both the technical and management perspectives.

The purpose of this paper, then, is twofold. On one level we attempt to show how educational technology may be intelligently applied to meet complex training requirements. From this technical perspective we describe the development of a sophisticated training system and how this system works to achieve an efficient learning environment. On a broader level, however, this paper describes how such a program must be managed in order to achieve this level of effectiveness in the practical world of cost and schedule constraints.

INTRODUCTION

This paper presents a training program being developed by General Electric for the United States Marine Corps and the U.S. Navy's Space Warfare Systems Command. This program's objective is to provide a "turn key" training system which will train U.S. Marine repairers and technicians to maintain the AN/TPS-59 radar. The paper describes a model program which emphasizes sound application of instructional theory, management principles, engineering applications, and development techniques. The subject-project also demonstrates the cost-effectiveness of using multimedia delivery systems in a sophisticated solid-state electronics maintenance training environment. These media include conventional classroom delivery, videotape, Computer-Based Instruction (CBI) cognitive materials, interactive graphic simulations (2D simulation), interactive videodisc, six high fidelity training device carrels, dummy equipment, and actual Prime Mission Equipment (PME). The ultimate advantage of the simulator is that it looks and responds like the actual equipment. This range of types of training achieves one of the major goals of the project: to provide the student with substantial hands-on practice time without endangering either the student or the radar equipment. This hands-on training is highly interactive with each scenario capable of delivery as a guided demonstration, process-remediated (step-level remediation) simulation, on product-graded, large-boundary, free-play exercise.

THE TRAINING PROBLEM

Recent trends in the procurement of sophisticated technology systems by the military have tended to concentrate equipment dollars and mission responsibility into relatively few major equipment systems. Electronics are being engineered for increasingly higher levels of reliability with self-correcting features; many systems are capable of self-diagnosis when faults occur.

When problems occur beyond the system's self-diagnosing capabilities, only highly trained technicians using rarely practiced skills can correct the faults. These systems, therefore, create an unparalleled demand for excellence and effectiveness of training. The complexity of the technology and of the corresponding knowledge base forces the equipment purchasers to use unacceptably long training pipelines unless innovative and highly effective training techniques are used.

The Gramm-Rudman Act will likely have a profound effect on training system procurement. This effect will primarily be focused through more stringent requirements to demonstrate positive cost benefits over the life cycle of the training system. Thus, future systems must be highly effective in teaching critical skills while simultaneously reducing training pipeline length and the concomittent life cycle costs of this training. The demand for cost-effective training programs that can keep pace with technology will continue to be a major issue through the turn of the century.

GENERAL ELECTRIC'S TOTAL TRAINING SOLUTION

GE's Educational Services Organization has developed an integrated instructional design system which quantitatively selects the most learning-efficient media for each objective, and integrates delivery in the context of a complete mastery-based training program. This program development system begins with an exhaustive front-end analysis and design phase, and concludes with two pilot courses and a plan for long-term evaluation and revision based on student performance on the job. A highlight of this system is the formal incorporation of cost/benefit analysis at several levels in the analysis, design, and development phases.

The integrated approach utilized here delivers the training for each job-relevant task with the most effective instructional technique for each objective. Highly critical tasks receive the most intensive instructional treatment through high fidelity, hands-on practice on a simulator. Less critical tasks, typically knowledge and skill instances which are common to many tasks, receive less cost-intensive instructional treatment such as group-paced instruction or cognitive CBI. Complete transfer of skills to the job environment is assured by practicing and testing job performance for all critical learning objectives using actual equipment.

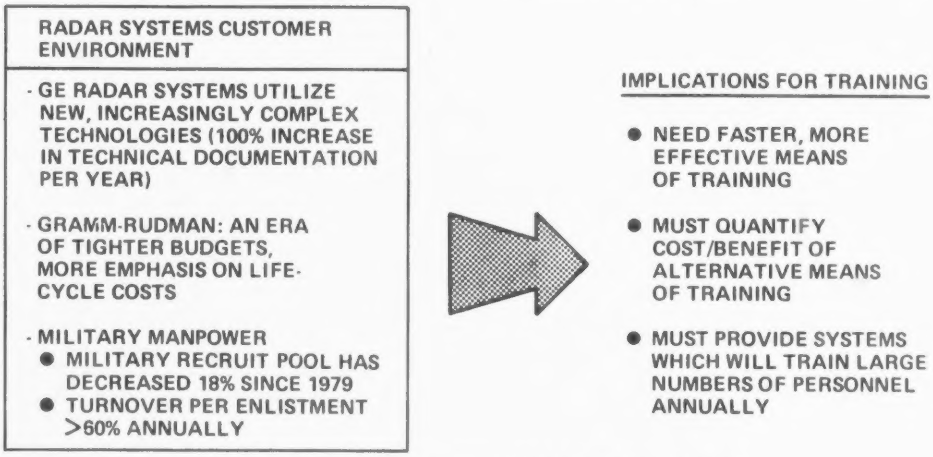

RADAR SYSTEMS CUSTOMER ENVIRONMENT
- GE RADAR SYSTEMS UTILIZE NEW, INCREASINGLY COMPLEX TECHNOLOGIES (100% INCREASE IN TECHNICAL DOCUMENTATION PER YEAR)
- GRAMM-RUDMAN: AN ERA OF TIGHTER BUDGETS, MORE EMPHASIS ON LIFE-CYCLE COSTS
- MILITARY MANPOWER
● MILITARY RECRUIT POOL HAS DECREASED 18% SINCE 1979
● TURNOVER PER ENLISTMENT >60% ANNUALLY
IMPLICATIONS FOR TRAINING
● NEED FASTER, MORE EFFECTIVE MEANS OF TRAINING
● MUST QUANTIFY COST/BENEFIT OF ALTERNATIVE MEANS OF TRAINING
● MUST PROVIDE SYSTEMS WHICH WILL TRAIN LARGE NUMBERS OF PERSONNEL ANNUALLY

FIGURE 1. EDUCATIONAL SERVICES CUSTOMER NEEDS

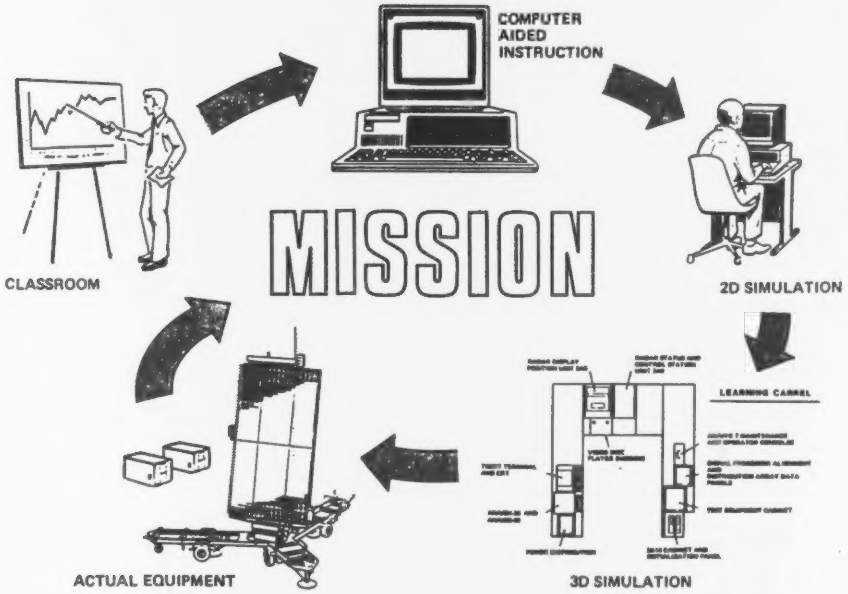


FIGURE 2. THE INTEGRATED INSTRUCTIONAL SYSTEM

A FORMULA FOR SUCCESS

General Electric uses a development approach designed for effectiveness and efficiency. The success of this program derives from five critical concepts together with a rigorous, disciplined commitment by management to build to the design specifications according to plan, within cost and schedule constraints.

The critical concepts are as follows:

- Formal application of cost/benefit analysis within a learning type strategy/media model.
- Hard-coupling engineering requirements to specific training requirements; instructional design, therefore, both drives and limits electrical and software designs for the training device.
- Incorporating a "living design" concept for course development and media selection.
- Defining a Problem Card Data Base System (PCDS) which uses specific PME fault parameters as specifications for program integration, cross-functional communication, and system test.
- Using a multidisciplinary team of specialists, including: instructional designers, Subject Matter Experts (SMEs), hardware and software engineers, graphics and CBI development specialists, and program planning and control specialists.

Each of these concepts entailed significant investments in front-end planning and management control over development. Each will be explained below, and the part each played will be described.

THE GE SYSTEMS APPROACH TO TRAINING METHOD

GE's unique Systems Approach to Training (SAT) is an evolution of the traditional Instructional Systems Development (ISD) approach used in most military training contracts (see Figure 3). The key to SAT is a first-pass task analysis and the early development of Terminal Learning Objectives (TLOs), which creates the minimum level of design detail necessary for a cost-benefit analysis of training alternatives. This activity (referred to as the Macro Design Phase) is documented to the customer as GE's Training Integration Plan (TIP). Detailed design specifications are developed after contracting parties agree to the direction

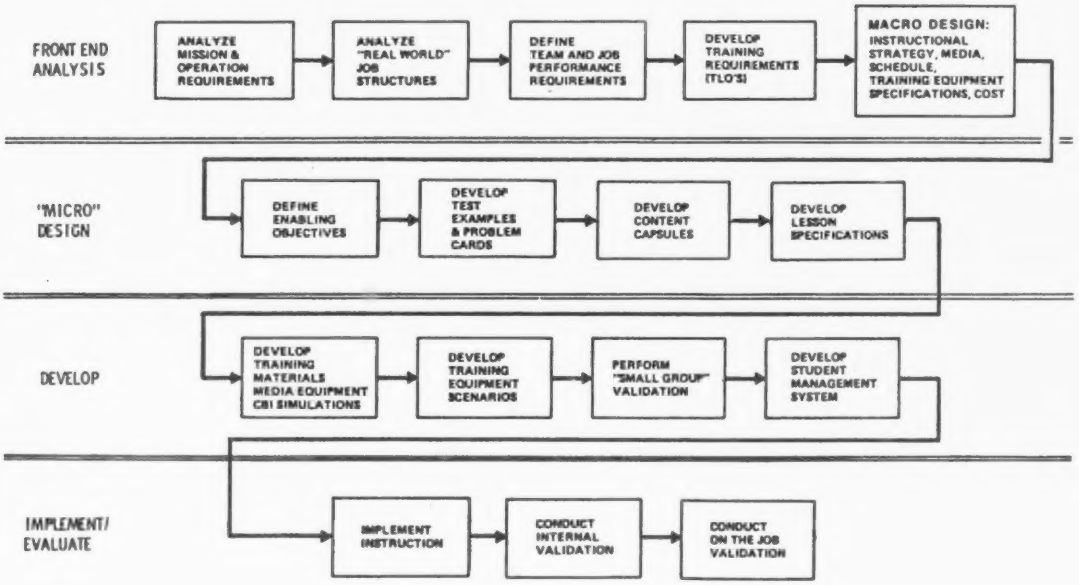

FRONT END ANALYSIS
ANALYZE MISSION & OPERATION REQUIREMENTS
ANALYZE "REAL WORLD" JOB STRUCTURES
DEFINE TEAM AND JOB PERFORMANCE REQUIREMENTS
DEVELOP TRAINING REQUIREMENTS (TLO'S)
MACRO DESIGN: INSTRUCTIONAL STRATEGY, MEDIA, SCHEDULE, TRAINING EQUIPMENT SPECIFICATIONS, COST
"MICRO" DESIGN
DEFINE ENABLING OBJECTIVES
DEVELOP TEST EXAMPLES & PROBLEM CARDS
DEVELOP CONTENT CAPSULES
DEVELOP LESSON SPECIFICATIONS
DEVELOP
DEVELOP TRAINING MATERIALS MEDIA EQUIPMENT CBI SIMULATIONS
DEVELOP TRAINING EQUIPMENT SCENARIOS
PERFORM "SMALL GROUP" VALIDATION
DEVELOP STUDENT MANAGEMENT SYSTEM
IMPLEMENT/ EVALUATE
IMPLEMENT INSTRUCTION
CONDUCT INTERNAL VALIDATION
CONDUCT ON THE JOB VALIDATION

FIGURE 3. THE SYSTEMS APPROACH TO TRAINING

proposed. This approach minimizes cost and schedule risk to the government client, as well as to the contractor.

This systematically planned program produced courses which cut training time by an average of 47 percent compared to a conventional training approach. GE's economic analysts showed that costs would be fully recovered after 6.5 years.

DESIGN & MEDIA SELECTION

A detailed design process (Micro Design) was required in order to precisely define all enabling objectives, prescribe an optimal media-mix, and sequence objectives to deliver the content. GE conducted a classic four-level task and skills hierarchical analysis of the job environment. Using the USMC unique ISD doctrine (MCDECO 1559.1A), and by contract agreement, GE extended the analysis down to two additional levels of prerequisite skills and knowledge required to accomplish each step. Common threads representing core skills and knowledges were identified and grouped into related enabling objectives. These "common core" objectives were sequenced at the beginning of the courses. These objectives, as well as all team activities, were designed as conventional group-paced instruction and related media.

The remaining critical steps, knowledge, and skills were rigorously analyzed and classified by learning level and type (see Figure 4) so that the most appropriate media could be chosen in each case. This also formed the basis upon which carefully defined sequences of skill-building objectives were formulated. These sequences would later form the basis for Problem Cards and simulation specifications. Finally, cost-benefit analysis in the development model allows instructional criticality to determine the cost ceiling of each particular portion of the training.

Beyond detailed analysis of the learning level, type, and cost ceiling for each objective, the following dimensions were also applied to determining each objective's media:

	FACT	VERBAL CHAIN	PROCEDURE	CONCEPT	RULE	PRINCIPLE
RECOGNIZE	F1	V1	D1	C1	R1	P1
RECALL	F2	V2	D2	C2	R2	P2
USE-AIDED	▨	V3	D3	C3	R3	P3
USE-UNAIDED	▨	V4	D4	C4	R4	P4

FIGURE 4. THE LEARNING TAXONOMY

CBI Cognitive

- To bridge the gap between general theory and application to instances
- To practice recalling generalities
- To relate cognitive information to procedures to be performed in interactive simulations
- To augment/support (not rewrite) technical manuals

2D Simulations, Interactive Graphics

- To build skills in equipment usage and procedural performance (part-task simulations)
- To provide immediate corrective feedback for predicted errors
- To provide drill and practice in critical part-tasks
- To train in multiple application areas (e.g., test equipment)
- To teach the functions of highly interactive single pieces of equipment (use of dials, switches, and indicators)
- To force the use of technical manuals

Videodisc

- To add realism in addition to fidelity
- To provide interactivity where graphics creation would be difficult or prohibitively costly (e.g., where depth perception is critical)
- To train in procedures where safety and space are factors
- To demonstrate team performances
- To elaborate on 2D simulations for demonstration or attention-focusing
- To model short motion sequences and psychomotor skills
- To add audio for emphatic value (safety issues) and to vary learning modality

3D Simulator

- To provide principle-level training through required student synthesis of specific procedures tailored to a particular class of faults
- To integrate part-task or partial skills into whole task performances
- To bridge/transfer 2D learning to actual equipment
- To provide substantial practice in critical tasks essential to long-term skill retention
- To force student into decision-making and application of previously learned knowledge and skills within a highly interactive environment
- To control student progress through increasingly complex performances while progressively decreasing aids, helps, and remediation (building self-reliance)
- To provide realistic training while reducing hazards to equipment and students
- To provide time for sufficient practice for tasks which require lengthy performances or a high quantity of performances

THE "LIVING DESIGN" CONCEPT

The development of the instructional design involved systematically specifying objectives and test items, selecting media and methods, and fine-tuning the instructional sequences. One of the most unique aspects of this approach was the formally planned overlap between design and production. This overlap enabled the implementation of a "living design" concept, wherein the design was refined and modified at the micro level to optimize instructional effectiveness without inflating costs. This flexibility meant the design had to be monitored and reviewed rigorously on an ongoing basis. However, the result was two carefully structured, precisely focused courses with a training system which meets all training requirements in the most cost-effective way possible.

The controlling tool used to provide the monitoring and review function for the design was the Lesson/Segment Specification developed by General Electric. This specification contains all the essential design information developed to date from the Micro Design, together with personnel assignments, development schedules, common error analysis, media utilization specifications, detailed interactivity specifications, administrative sign-off records, and other information. Changes to the formally frozen design required a documented request, review, and approval process through senior project staff to the program manager.

This approach compartmentalized cost and schedule risks at the "hour of instruction" (segment) level and completely avoided costly system-level redesign. This systematic interaction of the design and development process, in the long run, increased effectiveness of development dollars.

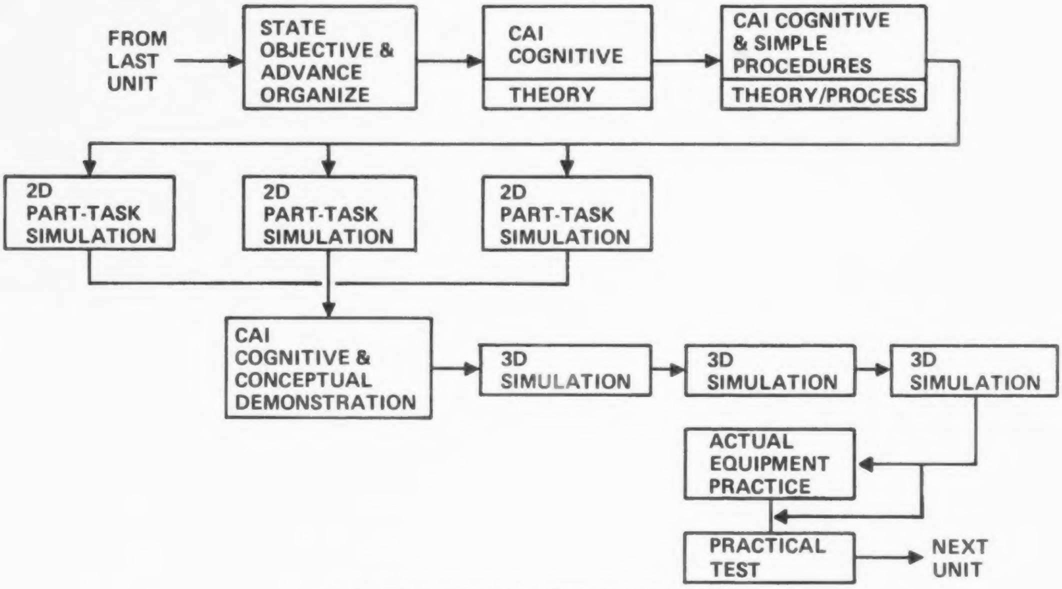

FROM LAST UNIT
STATE OBJECTIVE & ADVANCE ORGANIZE
CAI COGNITIVE
THEORY
CAI COGNITIVE & SIMPLE PROCEDURES
THEORY/PROCESS
2D PART-TASK SIMULATION
2D PART-TASK SIMULATION
2D PART-TASK SIMULATION
CAI COGNITIVE & CONCEPTUAL DEMONSTRATION
3D SIMULATION
3D SIMULATION
3D SIMULATION
ACTUAL EQUIPMENT PRACTICE
PRACTICAL TEST
NEXT UNIT

FIGURE 5. TYPICAL LESSON SEQUENCE

Figure 5 depicts a typical lesson sequence resulting from this approach. Each course contains at least one such sequence for each task required of the job incumbent. Thus, it can be seen that integration of many media had to occur at the task level, not just at the system or course level. The keys to achieving this level of integration within cost and schedule were the establishment of all training requirements, documenting those requirements in the form of Problem Cards, and allowing the design at the segment level to change on the basis of new information available only during the production process.

DEVELOPMENT

In order to produce courseware within time and cost schedules, the program was divided into projects corresponding to major media types after the Micro Design Phase provided the detailed specifications. Hardware and software design was carefully grounded in the training requirements through conversion to engineering requirements in the form of performance specifications. With these elements in place, all engineering work was also broken down and planned as projects. Development of the projects was then conducted in parallel with success contingent on adherence to the instructional design, engineering design, coordination, communication, and control between the development groups. Peak manning at critical points in this complex project hit highs of 353 personnel in a production-phase week.

SIMULATOR DESCRIPTION

The maintenance trainer was designed to support training in both manual and standard corrective maintenance, system theory, and system troubleshooting for the AN/TPS-59 radar. The trainer supports fault isolation tasks beyond the performance monitoring and fault location software provided as part of the radar. The simulator supports training at two skill levels referred to as repairer and technician levels, for two courses, averaging 19 weeks as a combined total.

The training device consists of an instructor station and six identical training carrels which closely resemble the radar. Each carrel functions as an independent device, training each student on that problem/task which matches that student's current needs. The carrel consists of a micro-VAX (TM) computer which runs all visible radar controls, indicators, and test points. All panels in the carrel are identical in form and function to those found in the radar. The heart of each carrel is a 10,000-pin backplane with which the technician student interacts using a real oscilloscope and digitally

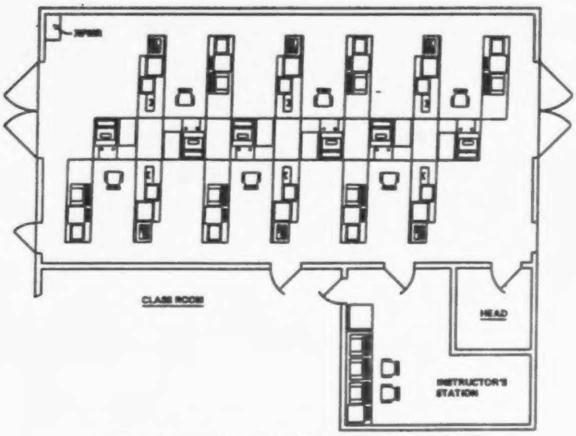

CLASS ROOM
HEAD
INSTRUCTOR'S STATION

FIGURE 6. TRAINING SYSTEM LAYOUT

recorded waveforms gathered from a prefaulted, actual radar.

Instructional system management is provided by a mature CBI system (the Hazeltine micro-TICCIT (TM) system) using GE-developed courseware and software. Each carrel contains, in addition to the radar-like equipment, a micro-TICCIT (TM) work station. Six standalone micro-TICCIT (TM) work stations are provided as well.

SIMULATION CONCEPTS: AN EXTENSION OF CBI INTO 3D HARDWARE

The networked CBI system controls the learning environment for student monitoring, feedback, cues, and grading. The simulator senses all student actions and provides high fidelity responses to these actions. Communication between the training device and the CBI system is activated only when author-specified actions, correct or incorrect, are emitted by the student. Thus, the device and CBI run each student's exercise in parallel, with the device providing the appropriate hardware response and CBI providing the appropriate instructional response such as scoring, feedback, prompting, logging, and remediation.

Thus, from the student's perspective, the device looks and acts like the real article in various faulty and operational states. However, each action is sensed, recorded, evaluated, and, if necessary, remediated.

The most signficant technical achievement of the GE program is the extension of the normal concept of CBI as a two-dimensional display system into what GE considers to be a fully-integrated CBI system. This integration was made possible through the use of a tool which GE terms the Problem Card Data Base System (PCDS). The Problem Card (PC) itself is simply a document which identifies the specific state of the equipment, under the conditions in which it is to be simulated in a particular exercise or scenario. Thus, a particular PC would contain the state of all lights and indicators, the effect of all controls, test points with associated signal parameters for correct and incorrect actions, etc. Thus the PCDS, is a major tool for program integration in that it acts as a basis for communication, a simulation specification, and as a system test specification.

To the Instructional Designer, a Problem Card is a strategy specification for simulation development. To the Test and Measurement person, it is a test item, instance, or entire instrument depending on its size, scope, and usage. All PCs designated for 3D simulation, taken as a group, define and limit the electrical, mechanical, and software requirements for a training device. To the Quality Control engineer, the same PC is a test specification. To the customer, Problem Cards give reality to a planned device and provide a firm basis for acceptance tests. Most importantly, to the student, each Problem Card is a new challenge which teaches and tests JOB BEHAVIORS.

In practice, a given PC is selected in the training environment based on each individual student's capabilities at his/her specific ability level at the moment. The level of cueing and feedback provided by any particular Problem Card is also conditioned by each student's current ability, the objective, and the learning modality being addressed (i.e., whether the

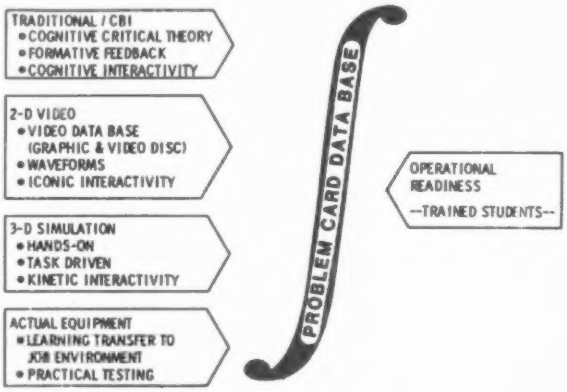

TRADITIONAL / CBI
• COGNITIVE CRITICAL THEORY
• FORMATIVE FEEDBACK
• COGNITIVE INTERACTIVITY
2-D VIDEO
• VIDEO DATA BASE (GRAPHIC & VIDEO DISC)
• WAVEFORMS
• ICONIC INTERACTIVITY
3-D SIMULATION
• HANDS-ON
• TASK DRIVEN
• KINETIC INTERACTIVITY
ACTUAL EQUIPMENT
• LEARNING TRANSFER TO JOB ENVIRONMENT
• PRACTICAL TESTING
PROBLEM CARD DATA BASE
OPERATIONAL READINESS
--TRAINED STUDENTS--

FIGURE 7. SYSTEM INTEGRATION

student is working through a guided demonstration, practicing, or testing). Thus, the PCDS assures that each student is always challenged to the limits of his/her ability at any point in time through the course.

Finally, the training system statistically checks each of its decisions against later student population performance and updates decision criteria based on these data. In effect, the system "learns" to make decisions which are instructionally better and better. These decisions are based on statistics maintained for each Problem Card and each student. This heuristic model implements an artificial intelligence strategy extending CBI into the 3D simulation realm and integrating the development activities of personnel as diverse as circuit card designers, courseware developers, and radar technician SMEs.

SUMMARY

Cost-effectiveness for this training system was determined in terms of shortened course lengths and life cycle student throughput, as well as the relative cost of providing hands-on training on actual equipment alone versus six training carrels and a total of twelve CBI terminals. The entire training program costs approximately the same amount as a single radar system dedicated to training purposes. Because the heart of the training system is software which operates on an author-defined, equipment-specific data base, the trainer's functionality may be easily modified or augmented in conjunction with long-term equipment changes as well as job environment changes.

Furthermore, by making what amounts to cosmetic changes to hardware equipment panels and development of the equipment-specific data base, entirely different electronic systems can now be simulated. This means that the basic electrical and software design need not be changed to simulate totally different Prime Mission Equipment. Development of these future training simulators will, thus, be at less cost and have shorter schedules than the pilot model.

Finally, the process model introduced through this program should contribute to substantially more effective training systems providing high fidelity/high interactivity hands-on training. The results obtained here can be generalized to other programs, regardless of training content, where educational technology can serve to reduce training time, enhance efficiency, and reduce life cycle costs.

Contact: Mike Olson or Harold Popplewell
General Electric Education. Serv.
EP-1, Room 130
P.O. Box 4840
Syracuse, NY 13221

315-456-3600

ADCIS Proceedings, November 1986

The Technology of Videodisc:

A Forecast From 1986 - 2001

G. Steven Tuthill

Courseware, Inc.

ABSTRACT

Videodisc hardware and software technology has advanced from a home entertainment device into diverse applications which include interactive simulation of hands-on training. Since the concept of videodisc was made possible, those who design, market, and implement opportunities for information access and dissemination have explored many applications for the technology. Applications of videodisc materials can be seen in educational settings, military environments, industrial settings, the consumer marketplace, and in public settings. While this technology opens many opportunities, its direction is not clear.

What will be the applications of videodisc technology and how will it impact society in the year 2001? Based upon the assumption that videodisc technology will have an ever increasing usage among many segments of society, the potential applications and impact of videodisc needs to be explored. This paper sets the stage and briefly describes the methodology of a study. It is the intent of this study to chart the possible future of videodisc technology to the year 2001. A Delphi Conference will be used to collect and characterize data from selected individuals representing a variety of fields. This paper was submitted prior to the actual study and outlines the rationale and architecture under which it is to be conducted.

Videodisc use is becoming widespread in many aspects of society. This media is gaining the attention of those who make decisions about the delivery methods of educational and training materials as well as information access. While videodisc has experienced growth and overall acceptance as a possible major communications media, both the near future applications and impacts need to be considered. Therefore, a study should be conducted which explores the considered opinions of individuals from various fields to identify the potential applications and impacts of videodisc upon specific areas and segments of society.

Hardware, software, and applications of videodisc programs have changed substantially since their entrance into the marketplace in the early 1970's. Videodisc technology has rapidly evolved in both sophistication and in usage. According to Weynand (1985), videodisc technology started with optical videodiscs, moved to capacitance electronic disc and finally evolved into the laserdiscs of today. New product and service announcements indicate that the technology has produced direct-read-after-write (DRAW) disc records which do not require a clean room environment, as well as write-once-read-many (WORM) systems which enables on-line writing of a program (Jaworski, 1986).

Early videodiscs were primarily developed for entertainment uses. Early developers explored the possibility of incorporating the videodisc in education and training settings. The developers found that Level I, linear, programs were inappropriate for education and training. Monies were made available by groups and foundations to further explore the possible applications of videodisc uses.

As a result, exploration of alternatives led to interfacing with computers to bring about the videodisc systems of today.

Videodisc systems are now used in education and training, communications, advertising, point of purchase, simulation, archieval storage, and information centers (Butler and Reeves, 1985). Videodisc programs are being developed to take advantage of advances in technology with the ability to involve the user as a source of input. The involvement of a user through an interface between various videodisc programs and a computer remains the subject of extensive research and development for applications of interactive videodisc.

According to Marx (1982), the technology of the videodisc has provided society with a learning tool that, when combined with the power of a personal computer, encompasses the collective attributes of every information device which has preceded its development. Videodisc applications contain media ranging from videotape to 35mm slide film, and involve teams of designers, actors, production personnel, and, for Level III, programmers. With the evolution of programming systems and tools, nonprogrammers have been able to directly create information sequences for videodisc. This development has also presented limitations. Rather than total creative freedom offered by languages, the designer must work within the parameters and abilities of the programming system. Research into productivity tools opens the possibility of expert systems to overcome these limitations, but add to research and development cost factors. Because of the numbers of people and sophistication of equipment required to produce videodiscs, it is an expensive communications media to produce. As cost effectiveness issues are met, videodisc applications should become more economically feasible and therefore accessible to a wider range of learners and consumers (Kleinmann, 1983).

Significant numbers of people are evaluating the potential for investment in videodisc equipment and programs in both home and commercial uses. While many positive developments have occurred, there are potential application problems.

According to Bork (1985), the combination of the computer and the videodisc should be able to produce the best of all possible worlds for educational use. However, Bork cites problems with the educational applications of the videodisc which include slow acceptance by learners, limited interfaces, limited use of the computer and little consideration of educational problems. Bork further states that technologists, rather than educators, tend to dominate production.

While technologists explore the addition of artificial intelligence in conjunction with the personal computer and the videodisc, potential uses are only beginning to emerge. What are the long-term possibilities of videodisc? Are the trends being established for videodisc the best or are there better uses for the technology? These questions and many like them were raised by Daynes in 1975 and are still relevant today. In an attempt to determine the potential effects of videodisc on education and training, a ten-year forecast study was conducted in the spring of 1976 by Daynes. His study used a Delphi methodology which, through a series of iterations, used the opinions of experts to predict the prospects for videodisc technology.

Results of Daynes' study showed the following:

- development of an industrial market separate from a consumer market
- videodisc formats shaped by the marketplace rather than influenced by government users or associations
- increased use of audiovisual materials in education and the use of videodisc technology for increasing instructional efficiency and reducing instructional costs.

Some forecasts which did not occur are:

- self-regenerating market producing enough videodisc business that vendors could survive among themselves
- acceptance of the technology by both consumer market and the education market
- 5% videodisc market saturation

According to Daynes (1986), the technology of the videodisc was still too new to obtain a clear picture of its future, as the experts of the day possessed more interest and enthusiasm about the media than experience. He believes that the particular forecasts which did not take place were prevented by many factors including changes in the technology, economic factors, trends in training, and the success of super-8 and video tapes.

Since Daynes' study, not only have there been technological advances in the industry, but also broader applications which have resulted in the exposure of videodisc technology to many segments of society. Further, videodisc technology will have an ever increasing usage as educators, trainers and manufacturers are currently developing this visual medium of knowledge transfer and information dissemination. For their efforts to be maximally productive, a current study needs to be conducted which explores the intended directions of the uses of videodisc technology. As a consequence, a study will be conducted to determine the potential applications and impact of videodisc technology during the period from 1986 to the year 2001.

This study plans to utilize the forecasting attributes of the Delphi methodology and a computer network for data collection. These instruments will be used to conduct a Delphi Conference with a series of three carefully planned iterations. These iterations are designed to gather, evaluate and characterize the various opinions of those with knowledge of the field. Based upon the final rankings of opinion, trends for future applications of videodisc technology and the impact of this media will be obtained. The study will emphasize future applications of videodisc technology and its potential impact upon various segments of society.

METHODOLOGY

The Delphi Method is not a strict instrument-based format. It is, according to Helmer (1975), concerned with the utilization of expert opinion and experts are rarely available for laboratory experimental studies. Therefore, those who have designed Delphi studies have adopted a Delphi philosophy, not a procedure. There are basically two aspects of the philosophy, the first is a study which is geared for consensus. This approach uses a heterogeneous group of individuals with the objective of obtaining a consensus. The second Delphi approach is that of a Policy Delphi. This approach uses a diverse group of individuals who are not considered expert, and do not have consensus as a goal. The participants are selected on the basis that their views are in opposition.

According to Turoff (1970), there are six phases in the communication process of a Policy Delphi. These phases are also used in the Delphi method and serve as guidelines for this study.

1. Formation of issues. What is the issue that really should be under consideration? How should it be stated?
2. Exposing the options. Given the issues, what are the policy options available?
3. Determining the initial positions on the issues. Which are the ones that everyone agrees upon and which are the unimportant ones to be discarded? Which are the ones exhibiting disagreement among the participants?
4. Exploring and obtaining the reasons for disagreement. What underlying assumptions, views or facts are being used by the individuals to support their respective positions?
5. Evaluating the underlying reasons. How does the group view the separate arguments used to defend various positions and how do they compare to one another on a relative basis?
6. Reevaluating the options. Reevaluation is based upon the views of the underlying "evidence" and the assessment of its relevance to each position taken.

Turoff states that this process could take five paper and pencil rounds. However, a three or four round procedure can be accomplished if the Delphi planners utilize the following:

1. the monitoring team devoting a considerable amount of time to carefully reformulating the obvious issues,
2. seeding the list with an initial range of options but allowing for the respondents to add to the list,
3. asking for positions on an item and underlying assumptions in the first round.

Turoff further states (1970:88), "Ultimately, the best vehicle for a Policy Delphi is a computerized version of the process in which the iterative round structure disappears and each of these phases for a given issue is carried through in a continuous process." The original design of the Delphi was intended to deal with technical topics and seek consensus among homogeneous groups of experts. This contrasts with the Policy Delphi in that it seeks to generate the strongest possible opposing views on the possible resolutions of a major policy issue. Turoff states that the Policy Delphi does not deal with experts, only informed advocates. Therefore, generation of consensus is not the goal of a Policy Delphi. It is a tool for analysis on policy issues.

Delphi studies dealing with planning problems have been conducted in government, business, and industry. The use of the Delphi technique as a forecasting instrument is documented in such studies as: The Future of the Telephone Industry; The Future of Newsprint, and the Nature of Economic Losses Arising from Computer Based Systems in the Next Fifteen Years. Today it is being used as a forecasting instrument in Western Europe, Eastern Europe and the Far East. An example of its application is a Delphi study concerning the medical industry which was conducted at the Philips Electric Corporation Plant in Eindhoven, Holland (Linstrone and Turoff, 1975).

This study intends to utilize the malleable format of the Delphi philosophy to form a hybrid design with the structure of a Delphi Policy and the homogeneous group consensus of the Delphi Technique. In order to have the greatest possible opportunity for accurately forecasting the future of the videodisc technology, strong views of diverse individuals need to be considered.

Subjects

Individuals who participate in this study will be recognized as being knowledgeable in their fields. These individuals will be asked to participate based upon the following criteria: publications concerning videodisc applications; involvement in the manufacture or application of videodiscs; memberships in professional organizations which have vested interests in the advancement of videodiscs; or are professionals who work with issues and the effects of videodisc technology. These individuals will represent divergent professions.

Apparatus

Delphi methodology consists of questions and data collection. A series of potential impact topics designed to stimulate thinking in specific areas are presented. The questions for this study were selected based upon the future of the technology in four basic categories: growth/development, impact, obstacles, and new technology. The combination of the Delphi technique with computer networking will enable individuals from widely dispersed geographic areas to participate in this study.

Questions

1. Characterize the growth of videodisc technology by the year 2001 in each of the following areas:
   A. Market saturation both industrial and home
   B. Applications of videodisc products
   C. The composition of those who produce videodiscs
   D. The composition of the videodisc products
   E. The production process
   F. The audience which uses the technology
   G. The production and consumer costs
   H. Other areas

2. Characterize the impact of videodisc technology by the year 2001 in each of the following areas:
   A. Training/education
   B. The general public
   C. Professions and jobs
   D. Communications
   E. Other technologies

3. What will be the obstacles to continued growth and development of videodisc technology?

4. What technology will follow videodisc?

Procedure

The study will take place in three parts:

Part One: A general question is asked relating to the future of videodisc technology. It is designed to elicit a number of predicted uses of the technology or its effect which each individual feels will occur over the duration of the forecast period. The results of the study will be predicated upon the responses to, "Based upon the expertise which you possess in your area of specialization, what do you expect to occur as a result of videodisc technology by the year 2001." Further, the participants will be provided with a series of areas to contemplate.

Part Two: Presents forecasts obtained from Part One to the individuals to rank in order of possibility within each grouped portion. They are also requested to place a confidence rating upon each of their responses. Individuals will then be asked to make statements about any of the predictions and add any additional predictions which they feel should be included.

Part Three: Individuals have a final opportunity to rank and provide a confidence rating for each of the remaining predictions. Statements of events which will not happen, based upon the statistical data, will be removed prior to final ranking and rating. Individuals will once again receive statistical data along with the field of choice. Additionally, the panel will be provided with ranking options for rounds two, and three such as: "probability (date)" of low ranking prediction statement, and "50-50" meaning an either way of potential occurence.

Analysis of data

The data gathered by the computer network is to be categorized by a Delphi committee composed of six individuals selected by the author. The data will be assessed using z scores. In addition, it will establish a correlation between the individual's ranking and Likert confidence rating. Comparisons will be made between and among the homogeneous groups.

Discussion

Videodisc use is becoming widespread in many aspects of society. This media is gaining the attention of those who make decisions about the delivery methods of educational and training materials as well as information access. Alfred Bork (1985), warns that the videodisc industry is driven by its technology and not those who design the software. Information derived from this study has the potential to inform those who design, manufacture or use the videodisc of its potential applications and impacts.

References

Bork, Alfred. 1985. "Why Has the Interactive Videodisc." The Videodisc Monitor, September, pp. 16-17.

Butler, Ronald W.; Reeves Perry. 1985. "Industrial Videodisc Applications: An Update" A paper presented at the Institute for Graphic Communication, Monterey, July 28-30.

Daynes, Rodney. 1977. "Videodisc Technology For Educational Use: A Ten Year Forecast." Master's Thesis Study. San Diego State University.

Daynes, Rodney. 1986. Personal interview, San Diego, 25 March.

Helmer, Olaf. 1975. The Delphi Method: Techniques and Applications. Foreward. Reading, Massachusetts: Addison-Wesley Publishing Company.

Jaworski, Joe. 1986. "Lasers Anticipate Mass (Storage) Appeal." Hardcopy, February, pp. 43-49.

Kleinmann, Jack. 1983. "Interactive Discs." Video Pro, February, p. 47.

Linstone, Harry A.; Turoff, Murrary, eds. 1975. The Delphi Method Techniques and Applications. Reading, Massachusetts: Addison-Wesley Publishing Company.

Marx, Rayond J. 1982 "Videodisc-based training: Does it make economic sense?" Training/HRD, March, p. 56.

Weynard, Diana. 1985 "Laserdiscs: History and Applications." EITV, August, pp. 46-48.

Contact: G. Steven Tuthill
10075 Carroll Canyon
San Diego, CA 92131

619-578-1700

USING COMPUTER DISPLAYED PICTURES TO ENHANCE VISUO-SPATIAL APTITUDE IN PRE-ADOLESCENT FEMALES: SUGGESTIONS AND OPPORTUNITIES FOR RESEARCH

Patti R. Baker - The Ohio State University

John C. Belland - The Ohio State University

Abstract

The underutilization of women in careers built upon mathematics and science is seen in part as a function of women's lack of the visuo-spatial ability necessary to understand and interpret scientific and mathematical phenomena. New technologies provide unique opportunities to investigate the development of visual-spatial aptitude. There is substantial evidence that visuo-spatial aptitudes can be taught, but research and intervention traditionally occur after females already lag behind the spatial development of males. A more timely and productive intervention strategy may be to provide visuo-spatial before adolescence, a point at which spatial aptitude differences between males and females begin to appear. This paper presents the rationale and strategies of research on the visuo-spatial development of pre-adolescent females and how the research can accommodate the regular curriculum with new technologies yet provide a foundation for a meaningful, long-term research agenda.

INTRODUCTION

Visuo-spatial ability is necessary for the understanding of geometric concepts, the interpretation of engineering drawings and the understanding of multidimensional concepts (Bohm, 1981; Lowry and Knirk, 1982; Lantz, Carlberg, and Eaton, 1982). This ability interacts with a person's being able to understand pictorial information in any medium including computer screen displays. In addition, this ability interacts with a person's being able to mentally manipulate either displayed pictorial information or self-induced images.

Although an examination of the literature shows conflicting evidence as to the amount, time of onset, and causes of differences in visuo-spatial aptitude, there is a consensus that most females do not achieve the same level of spatial skills as males. Various studies over several decades point to the propensity of females to score lower on spatial tests than males (Barratt, 1955; Fennema and Sherman, 1977; West, 1984). At the same time, sex-related differences in science and mathematics have also been documented (Armstrong, 1980; Pfafflin, 1984) in that boys tend to produce higher achievement scores than girls. Other studies have shown that persons with high spatial abilities have higher achievement and more frequent participation in mathematics (Lowery and Knirk, 1982) and in science and engineering (Lantz, Carlberg, and Eaton, 1982). Consequently, the mathematics achievers, scientists, and engineers are more frequently males than females.

This inequality in spatial abilities has far-reaching consequences. Boys decidedly outnumber girls in advanced mathematics courses in high schools. NCES 1980 statistics show that female participation on high school algebra and geometry classes was about even, but males outnumbered females in trigonometry classes by a third and by 2 to 1 in calculus courses (Buccino, Evans, & Tressel, 1982:59). Furthermore, even though total enrollment of women in American institutions of higher education is over 50% (Dearman & Plisko, 1981: 146), women make up only 13% of the undergraduates in engineering and 29% in physical science. Women enrolled in engineering and physical science graduate programs number comprise only 7% and 18%, respectively. In the period charted from 1970 to 1980, the number of women earning bachelor's degrees in mathematics actually declined (Buccino, Evans, & Tressel, 1982:71).

The implications of such findings are serious. There is an underutilization of half the population in careers that are built upon mathematics and the sciences. Not only is a great portion of the population not enjoying the benefits of technology jobs that typically command higher salaries, but the United States is not utilizing a segment of its population that might contribute to scientific innovation and productivity.

What Can Be Done?

There is evidence that visuo-spatial aptitudes can be taught (Bishop, 1982; Lowery and Knirk, 1982; West, 1984; Zavotka, 1985), especially with computer-based activities. Bishop (1980) reported on the successful use of microcomputer video games resulted in spatial visualization acquisition. Research and intervention, however, traditionally occur after females already lag behind the spatial development of males even though a remedial approach can be more costly and less effective than a proactive approach. A more timely and productive strategy may be to identify critical intervention points so that the visuo-spatial aptitude of females can be maintained at parity with that of males.

The appropriate time for intervention seems to be pre-adolescence. Although spatial skills are shown to be improved with training, the timing of intervention appears to be crucial (Bishop, 1980). A study by Smith and Schroeder (1979) found that fourth graders generally benefited from spatial skills instruction but that there were no significant differences in post-instruction performance between boys and girls. Smith and Litman (1979) replicated the study with 11-13 old students in order to determine whether there were differences in spatial visualization ability among early adolescent students and whether instructions would affect boys and girls differently. The investigators found that only the boys benefited from instruction. By combining the results of this study with the results of the earlier Smith and Schroeder study, the authors concluded that girls and boys in the fourth grade benefit from spatial instruction, but instruction postponed to early adolescence benefits only the boys. Timing of intervention, they concluded, is crucial; pre-puberty may be an optimum time for spatial instruction.

In addition to the timing of the intervention, the type and structure of activities used for intervention are also important elements of an effective approach to developing visuo-spatial skills in females. Hawkins (1984) suggests that the nature of instructional tasks used to teach mathematics concepts is more likely to engage boys than girls. If so, it will be essential to develop tasks which appeal to girls so that they will be naturally inclined to develop spatial skills and pursue mathematical ideas more intensely and extensively. Bishop (1980) posits that focusing upon primary abilities such as spatial aptitude would develop mathematical ability more effectively than just emphasizing mathematics content and skills.

High resolution computer generated three dimensional animation and interactive high resolution computer graphics offer new tools for developing spatial aptitude. Unlimited perspectives, movement through a mathematically defined three dimensional space, and ease of manipulating design elements set computer generated animation apart from other media (see Figure 1). Interactive computer graphics allow the learner to make and test spatial hypotheses. Although Piaget (1971:74) suspected the usefulness of visually mediated learning as compared to the manipulation of

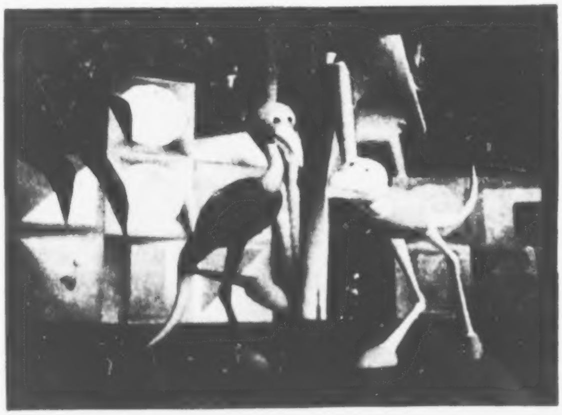


Figure 1

objects, the research of Olson and Bialystok (1983:164-181) presents evidence that direct block manipulation training and visually mediated experiences in block rotations both provided significant results. Furthermore, Salomon (1979) repeatedly documents the effectiveness of modeling spatial representations (see Figure 2). The superiority of one mode of learning over another, however, does not need to be the focus of research as much as the investigation into the effects of learning from types of images that have never before been available. These new tools allow the modeling of a mental skill that doesn't naturally occur by using the media to supplant that skill.

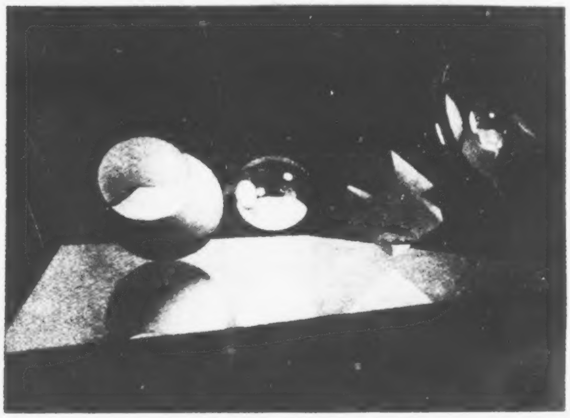


Figure 2

Research Objectives for the Development of Visuo-Spatial Skills

Sensitively designed research and development activities would focus on the following objectives:

- To identify and/or develop instructional experiences which hold promise for increasing the spatial skills of pre-adolescent females.
- To identify various combinations and sequences of those instructional experiences which develop visuo-spatial skills.
- To determine the cognitive styles of pre-adolescent females as a basis for designing appropriate activities for spatial skills training.
- To link spatial skills training with regular school curriculum including design of experiences for teachers (usually female) to help them develop their own spatial skills.

Long term research goals would additionally include the following objectives:

- To analyze the long-range effects of spatial skills training through longitudinal research.
- To plan a large-scale implementation of effective spatial skill training sequences.
- To investigate the applications of the developed activities and software within and between groups (e.g. females of different socio-economic status, pre-adolescent males and females).

Investigations

To meet the shorter term objectives, investigations would consist of selecting and writing appropriate software for spatial skills instruction, implementation and testing of instructional activities, and revision and retesting.

Given present technologies, software is suggested to consist of three types: pre-existing computer languages, locally produced software utilities (programs that utilize and extend the capabilities of the existing software), and computer generated animation. The software, to be used individually and in various combinations, is suggested to be selected and designed to appeal to the interests and motivations of pre-adolescent young women and will be linked to curricular and extra-curricular activities. Appeal factors will be identified via observation and interviews with teachers and students both previous to and during use of the project materials and activities.

Several software packages are already available that are appropriate for the development of spatial skills on the Apple Macintosh or the Apple IIe. Identified programs for the Apple IIe include "Funky Chicken" developed by the TABS lab at The Ohio State University and currently distributed by Encyclopedia Britannica. Software developed by Sunburst also appears to be useful: "Tip 'n Flip", "The Factory", and "Superfactory". Students using "Tip 'n Flip", for instance, mentally flip a pattern vertically or horizontally until a pattern matches the specified design. Enabling Technologies has developed software for the Macintosh named "Easy 3-D", an advanced hidden surface modeling software for artists, architects, engineers, and designers. A computer language also worthy of consideration is Experintelligence's "ExperLOGO", a Macintosh version of the programming language LOGO developed by Seymour Papert at MIT. ExperLOGO allows the user to explore mathematical concepts via the computer and to develop visual models of mathematical constructs in three dimensional space. Activities to be developed and refined would enable young women to use these computer programs to improve spatial skills. Users of Easy 3-D, for instance, are able to render drawings of three dimensional objects on the video display and are then able through keyboard commands to rotate that object on the horizontal, vertical, and/or depth axes as well as change eyepoint or lighting perspective (see Figure 3).

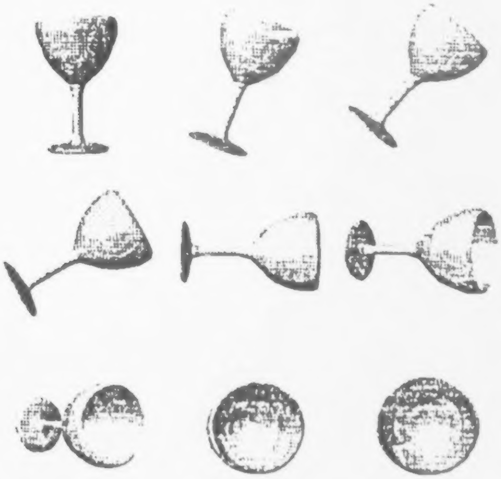


Figure 3

In addition to ExperLOGO, software utilities could be designed and programmed for the Macintosh that would enhance, extend, or supplement the ExperLOGO activities. Some of these utilities would use computer animation techniques on the Macintosh. The Macintosh is an extremely user-friendly and powerful computer that has scarcely been utilized in education as yet, but promises to be an excellent instructional tool. Both the availability of innovative software and the appeal of the machine and its operating system make the Macintosh an excellent tool for providing pre-adolescent girls with appropriate spatial development activities.

Three sets of criteria should be considered for the selection and development of activities:

1) Specific spatial skills must impact strongly on the acquisition of specific mathematics and science concepts in high school and college curricula.

- Spatial skills should enable the interpretation of various forms of scientific, technical and mathematical drawings; for example, orthographic projections (plans, elevations, and end views), cross-sectional drawings, perspective line drawings, and diagrammatic models.
- Spatial skills should enable the imagining of phenomena as they would appear from different angles or at different times.
- Spatial skills should enable simultaneous thinking about multidimensional phenomena.
- Spatial skills should enable not only the mental manipulation of objects or phenomena but also enable the orienting of oneself in three-dimensional space.

2) Spatial skill activities must have strong appeal to girls while taking care to avoid the reinforcement of limiting cultural stereotypes.

- Spatial skills activities should emerge from either the expressed interests of intermediate-grade girls or should be drawn from observable elements in their culture.
- Spatial skills activities should vary depending on socio-cultural milieu.

3) Spatial skill activities must integrate with ordinary intermediate grade curriculum.

- Spatial skills activities should in-so-far-as-possible be designed to substitute for regular activities in the curriculum instead of being added on. In other words, spatial skills activities should show promise for accomplishing regular curricular objectives such as understanding simple machines more efficiently and effectively.

- Spatial skills activities should have a high degree of intrinsic motivation and should be perceived as "important stuff" by students, teachers and parents.

Another important consideration of classroom-based research is the identification of the participating classroom teachers. Although careful identification and planned participation of classroom teachers should always be considered, it is important for this suggested research in two regards. One, although it is important to construct materials that help learners learn, it is also imperative to construct learning materials to help teachers teach. The classroom teacher is an important link between development and implementation and is a source whose perceptions of the utility and effectiveness of instructional materials is not only to be understood but to be sought. Secondly, since most elementary teachers are women, and women tend to have underdeveloped spatial skills, the teacher is likely to be as spatially untrained as the students. This deficiency not only impedes the teacher's own understanding of the usefulness of spatial skills training, but also has inhibited understanding of mathematical and scientific phenomena. Such inhibitions are almost inevitably passed along to students.

Conclusions

Researchers need to investigate further how spatial skills inequality between the sexes can be eliminated or reduced by educational experiences. Not all women need to pursue advanced mathematics or be technologically trained, but they should be given opportunities to develop such skills. The goal for educators should be to prepare children for options in careers, not destinies that are determined by lack of training or opportunity. Keeping young women in the mathematics and science classes in high school increases the pool of potential engineers and scientists (Matyas, 1985). The future well-being of the American society depends upon a working populace that is prepared to use technology.

New computer technologies not only provide unique opportunities for the modeling and training of visuo-spatial skills but also confront researchers and practitioners with a multitude of questions regarding the comprehension of iconic information, static or animated, displayed via computer terminals. New directions are needed in visuo-spatial research if the results of such research are going to be used to develop ability rather than to merely remediate deficient skills. The nature of the tasks used for skill development also need to use the newer technologies to address the cognitive, social, and affective needs of the target population. The availability of new technologies provides unique opportunities for investigation of visually mediated learning and the development of visuo-spatial aptitude.

Bibliography of Pertinent Literature

Armstrong, Jane M. (1980). Achievement and participation of women in mathematics: An overview. ERIC Document ED 184 878.

Baker, P. R. (1985). State-of-the-art computer generated animation and the emerging curriculum: Teachers' perceptions of possibilities. Unpublished manuscript.

Barratt, E. S. (1955). The space-visualization factors related to temperamental traits. *Journal of Psychology*, *39*, 279-287.

Bishop, A. J. (1980). Spatial abilities and mathematics education - A review. *Educational Studies in Mathematics*, *11*, 257-269.

Bishop, A.J. (1973). The use of structural apparatus and spatial ability - a possible relationship. *Research in Education*, *9*, 43-49.

Bohm, D. (1981). Insight, knowledge, science and human values. *Teachers' College Record*, *82*(3), 308-402.

Buccino, A., Evans, P. & Tressel, G. (1982). *Science and engineering education: Data and information*. Washington, DC: National Science Foundation.

Cambre, M. A., Taylor, W. & Belland, J. (1983). The effects of message design variables on the appeal and comprehension of computer generated instructional animations. Final Report: Grants to Young Scholars, The Ohio State University.

Cambre, M., Baker, P., & Belland, J. (1986). Effects of manipulating transparency/opacity of objects on learning the concept of the relationship of edge to volume of a cube. Paper presented at the Annual Meeting of the American Educational Research Association, San Francisco, April 20.

Dearman, N. B. & Plisko, V. W. (1981). The condition of education: 1981 Edition. Washington, DC: National Center for Education Statistics.

Fennema, E. and Sherman, J. (1977). Sex-related differences in mathematics achievement, spatial visualization and affective factors. American Educational Research Association Journal, 14 (1), 57-71.

Fennema, E. and Tartre, L. A. (1985). The use of spatial visualization in mathematics by girls and boys. Journal for Research in Mathematics Education, 16(3), 184-206.

Hawkins, J. (1984). Computers and girls: Rethinking the issues (Report No. 24). New York: Bank Street College of Education.

Lantz, A., Carlberg, C., and Eaton, V. (1982). Women's choice of science as a career (National Science Foundation Grant SED 8020854). Denver, CO: E.S.R. Associates.

Lowery, B.R. and Knirk, F.G. (1982). Micro-computer video games and spatial visualization acquisition. Journal of Educational Technology Systems, 11(2), 155-166.

Matyas, M.S. (1985). Keeping girls in the science track. Curriculum Review, 24(3), 75-78.

McKenna, F.P. (1984). Measures of field dependence: Cognitive style or cognitive ability? Journal of Personality and Social Psychology, 47(3), 593-603.

Naour, P. (1985). Brain/behavior relationships, gender differences, and the learning disabled. Theory Into Practice, 24, 100-105.

Pfafflin, S. M. (1984). Women, science, and technology. American Psychologist, 39, 1183-1186.

Smith, W. S. & Litman, C. I. (1979). Early adolescent girls' and boys' learning of a spatial visualization skill. Science Education, 63, 671-676.

Smith, & Schroeder (1979). Early adolescent girls' and boys' learning of spatial visualization skills. Science Education, 63, 61-66.

Skolnick, J., Langbort, C., and Day, L. (1982). How to encourage girls in math and science. Englewood Cliffs, NJ: Prentice-Hall.

Tobias, S. (1978). Overcoming math anxiety. New York: W.W. Norton.

West, C.E. (1985). Enhancing mathematics ability in sixth grade females via computer based graphics and problem solving (Doctoral dissertation, University of Southern California, 1984). Dissertation Abstracts International, 45, 1985A.

Zavotka, S. (1985). The use of three-dimensional computer graphics animation to teach spatial skills to home economics college students. Unpublished doctoral dissertation, The Ohio State University.

ADCIS Proceedings, November 1986

The Role of Computer-Assisted Instruction in Mastery Learning

David W. Dalton

Indiana University

ABSTRACT

There is considerable evidence to suggest that CAI is an effective instructional delivery system in many situations. Cross-age tutoring, like CAI, can also be a relatively powerful instructional improvement intervention. This study examined the effects of combining the benefits of cross-age tutoring with conventional tutorial CAI by comparing the learning outcomes from five instructional strategies: conventional instruction, mastery instruction, mastery instruction with CAI remediation, CAI only, and CAI with tutoring. Results of the study indicate that the instructional treatments which employ tutorial remediation strategies are more effective than those that do not.

INTRODUCTION

In the past decade, research has shown that computer-based instructional programs have been effective in improving learner performance and attitudes towards instruction.

For example, Kulik, Bangert, and Williams (1983) suggest that typical computer-based instructional programs produce a gain of .5 standard deviations over similar "conventional" instructional programs. In a recent follow-up analysis, Kulik, Kulik, and Bangert-Drowns (1985) note similar results when computer-assisted instruction (CAI) programs are used with elementary learners.

However, as Clark (1985) points-out, many of the studies reported in these analysies contain highly confounded results. Clark asserts that once competing explanations such as novelty, length of instructional time, and teacher factors have been adequately controlled, there is little if any effect directly attributable to the computer delivery system.

In another recent review comparing CAI with three other interventions designed to improve instruction, Levin, Glass, and Meister (1986) note that CAI, although more effective than moderate reductions in class size and increasing instructional time, is far less effective than cross-age tutorial programs.

One of the principal alledged benefits of CAI as a delivery systems lies in its ability to provide consistent and differentiated learner feedback (Clement, 1981).

It seems plausible that CAI could prove to be a useful medium for providing tutorial remediation in mastery learning settings since the quality of the feedback delivered through CAI can provide learners with specific information on their errors and help improve their performance. Although this proposition seems logical, more research is required to determine the precise role of CAI delivery systems in mastery learning environments.

This study reports the effects of CAI and cross-age tutoring systems on learner achievement and attitudes towards instruction in a mastery learning context.

METHODS

Subjects

The subjects chosen for partcipation in this study were 117 eighth grade learners selected from five sections of an eighth grade mathematics class. The sample consisted of approximately equal numbers of males and females and was composed primarily of Anglo students, with some minority learners.

Materials

The basic instruction for this study consisted of a lesson on computing the area of a circle. The lesson was divided into three major sections. The first section was an introduction to the concept "circle" and the number pi. The next lesson section described how to calculate the area of a circle when given the radius. The final lesson segment described how to calculate the area of a circle when the diameter was given. The basic lesson was modified into the following five parallel forms.

Mastery instruction. In this treatment, learners were given three five minute lectures on the three topics described above: the concept circle and the number pi, finding the area of the circle given the radius, and finding the area of the circle given the diameter. After the second and third lectures, the learners were given approximately 10 minutes to solve a series of ten short problems pertaining to the content of the lecture. At the end of the second problem set, the learners were given a 10-item quiz designed to measure their mastery of the skill. The quizzes were then graded and the learners were categorized by their mastery of the learning material. The following day, the students at or above mastery level (70%) were directed to complete an unrelated basic skills assignment. Learners below the mastery level were given an additional set of 10 problems and were tutored by their teacher on the problem solutions. At the end of this class period, all students were given a 20-item mastery test.

Mastery instruction with CAI. In this treatment, the learners were given the lectures as described above, followed by two 10-item problem sets, and a 10-item quiz. On the following day, learners not at the specified mastery level (70%), were given a CAI drill-and-practice lesson. This lesson consisted of the same 10 items found on the remedial worksheets provided to the learners in the mastery instruction treatment. The learners were shown the problem and provided with time to input the correct numerical answer. If the learner inputted the correct answer, the computer would display a phraise such as, "Good. The correct answer is..." followed by the correct answer. If the learners typed the incorrect answer, the computer would provide differential feedback. For example, if the learner used the incorrect formula, the computer would remind them of the correct formula. For each item, four potential errors were anticipated and feedback was designed for each of these anticipated responses. If the response was unexpected, the learners were told that the answer was incorrect and the correct method of solution was reviewed. In all cases, learners were then given the opportunity to try the item again. As in the Mastery instruction treatment, learners at or above the mastery level were given an unrelated basic skills worksheet. At the end of the class session, all learners were given a 20-item mastery posttest.

CAI lesson. In this treatment, learners were given a CAI tutorial lesson on the basic concepts. The tutorial was followed by problem sets identical to the ones employed in the two treatments previously described. At the end of the CAI lesson, learners were given a print-based quiz to assess their current mastery level. The following day, learners not at mastery level were given the CAI drill-and-practice program previously described, while learners at the mastery level were given the same unrelated worksheet. At the end of the class period, all learners were given the 20-item posttest.

CAI with tutoring. This treatment was identical to the CAI treatment described with the following exception. During the second day of the program, learners not at the mastery level were given a print-based worksheet and tutored by their teacher. Again, learners at mastery were given an unrelated basic skills worksheet and all learners were given the 20-item posttest at the end of the class session.

Control. This treatment represented conventional instruction. Here, all learners were given the three lectures, followed by two 10-item problem sets. At the end of the first period, all learners were given the 10-item mastery quiz. The following day, all learners completed the additional practice worksheet independently. At the end of the period, all learners were given the 20-item mastery posttest.

Dependent Measures

After the first day of their respective treatments, all learners were given a print-based mastery quiz on computing the area of a circle. This quiz contained five short-answer problems that concerned finding the area of a circle given the radius and five short-answer problems concerning finding the area of a circle given its diameter. Reliability of this quiz was .77 using split-half data obtained during the study. Results on this measure were used only to determine the learners' mastery level prior to assigning remedial help.

Mastery posttest. At the conclusion of the study, all learners were given a print-based 20-item mastery posttest, consisting of 10 radius and 10 diameter short-answer problems. Split-half reliability of this test was .96 using data obtained during the study.

Attitude survey. In addition to the achievement measures described above, the learners were also given an annonymous print-based twenty-item Likert-type survey designed to assess their attitudes towards the instruction. The split-half reliability coefficient of this instrument was found to be .87, again, using data obtained during the study.

Procedures

Learners were randomly assigned to their respective treatments where they completed the instruction. Immediately following the lesson, the learners were given the mastery posttest and the attitude survey previously described.

Data Analysis

Scores on the mastery posttest were analyzed through fixed-effects ANCOVA procedures, using sixth grade Comprehensive Test of Basic Skills (CTBS) total mathematics scores as the covariate. Attitude data from this study was analyzed with fixed-effects ANOVA procedures. Multiple comparisons for both datasets were performed using the Neuman-Keuls procedure.

RESULTS

The results for the mastery posttest are contained in Table 1. The adjusted means of the Mastery instruction with CAI treatment, the CAI with tutoring treatment, the Mastery instruction treatment, the CAI treatment, and the control groups were 91.13%, 88.17%, 85.71%, 75.23%, and 64.61%, respectively. These means were significantly different ($p < .001$). Specifically, the mean of the Mastery instruction with CAI group was significantly greater than the means of the control ($p < .001$) and the CAI ($.001 < p\ .01$) groups. However, the means of the Mastery instruction with CAI, CAI with tutoring, and Mastery instruction groups were not statistically different ($p > .05$).

The mean of the CAI with tutoring group was significantly larger than the control ($p < .001$) but not the Mastery instruction or CAI groups ($p > .05$). The mean of the Mastery instruction group was statistically greater than the mean of the control ($.001 < p < .01$), but not the CAI with tutoring group ($p > .05$).

Finally, the mean of the CAI group was not statistically different than the mean of the control ($p > .05$).

The results for the attitude survey are also contained in Table 1. The means of the Mastery instruction with CAI treatment, the CAI with tutoring treatment, the Mastery instruction treatment, the CAI treatment, and the control groups were 71.83%, 71.04%, 68.00%, 64.63%, 55.43%, respectively. In this case, the means of all the non-conventional interventions, Mastery instruction with CAI treatment, the CAI with tutoring treatment, the Mastery instruction treatment, the CAI treatment, differed significantly with that of the Control ($.01 < p < .05$), but none of these means were themselves significantly different ($p > .05$).

Table 1. Mean percent scores for the mastery posttest and attitude survey.

	Mastery Posttest	Attitude Survey
Mastery with CAI	91.13%	71.83%
CAI with tutoring	88.17%	71.04%
Mastery instruction	85.71%	68.00%
CAI	75.23%	64.63%
Control	64.61%	55.43%

DISCUSSION

There are three major results from this study that warrant further discussion. First, the three treatments that contained a tutoring component, whether deliverd by CAI or by the instructor, were significantly more effective than those treatments which contained no tutoring. This result supports the findings of Levin, Glass, and Meister (1984) that suggest that cross-age tutoring, in this case, teacher-pupil tutoring, can be a powerful instructional intervention. Since the principal focus of these tutoring efforts was in providing accurate and consistent feedback to the learners, it seems plausible that the differences observed between the tutoring and non-tutoring treatment groups is largely attributable to the presence of this feedback.

Second, Clark´s (1985) assertions that the effects of a particular delivery system often pale when compared with the effects of the delivery strategy were also supported by this study. Specifically, there was no significant differences between CAI and teacher-led treatments, as long as tutoring was included.

The reader should be note at this point that the teacher variable was controlled in this study since the same teacher designed and delivered all lessons. In addition, instructional time was controlled to the extent possible. Obviously, different learners would read and respond to the CAI tutorial at different rates, as well as requiring varying amounts of help during tutorial sessions. However, each treatment group was given exactly two instructional periods to complete their respective lesson materials.

Finally, the only significant differences observed on the attitude scale came between the control group and each of the other treatments, while none of the non-control treatments differed significantly. This result, coupled with observations made during the study, suggests that learners respond favorably to the individual attention and improved success provided them by the mastery learning treatments, in addition to the novelty of a non-traditional educational environment.

The reaffirmation of the importance of the instructional design and delivery strategy suggested by the results of this study should not be misconstrued as a condemnation of CAI as an instructional delivery system. Instead, this result suggests that the essential ingredient for successful instruction is the instruction itself and not the delivery vehicle; it is the "message, not the medium."

One of the principal benefits of CAI and other forms of stand-alone programmed instructional strategies lies in the necessity for developing the lesson content systematically and carefully. In short, the design process that accompanies the development of many CAI lessons offers a significant improvement over conventional approaches to instruction.

The results of this study may lead some to conclude that well-designed teacher-led mastery programs, by virtue of their lower cost, might be preferrable to the use of CAI. Unfortunately this conclusion neglects the added expense and time required for the development of the instructional materials, as well as the additional training necessary for this effective teacher-led to take place in the conventional public school environment. Clearly, more detailed analyses should be undertaken before meaningful cost/benefit judgements can be made regarding the use of CAI.

In summary, the results from this experiment suggest that tutoring can be a powerful supplement to both conventional mastery-based and computer-assisted instruction. Future researchers should focus on ways in which tutoring programs can be incorporated into mastery-based instructional programs. Since CAI provides individualized and differentiated feedback, further research may provide information on how this technology can be used to efficiently and effectively provide this tutoring.

REFERENCES

Clark, R. (1985). Confounding in educational computing research. Journal of Educational Computing Research, 1(2), 137-148.

Clement, F. (1981). Affective considerations in computer-based education. Educational Technology, 21(4), 28-32.

Kulik, J., Bangert, R., and Williams, G. (1983). Effects of computer-based teaching on secondary school students. Journal of Educational Psychology, 75(1), 19-26.

Kulik, C., Kulik, J., and Bangert-Drowns, R. (1985). The importance of outcomes: A reply to Clark. Journal of Educational Computing Research, 4(2), 381-87.

Levin, H., Glass, G., and Meister, M. (1984). Cost-effectiveness of four educational interventions. Stanford University, CA: Institute for Research on Educational Finance and Governance.

ADCIS Proceedings, November 1986

A MICROCOMPUTER SYSTEM FOR DIAGNOSTIC TESTING OF COGNITIVE ERRORS

John M. Eddins

University of Illinois at Urbana-Champaign

ABSTRACT

Under sponsorship of the Office of Naval Research, Kikumi Tstsuoka at the University of Illinois has developed a measurement model called Rule Space, which can be applied to computerized tests to diagnose students' cognitive errors. Application of this technique to a training environment is now being attempted at the San Diego Naval Training Center. A flexible test administration system using microcomputers was installed, capable of administering regular as well as adaptive tests, of collecting response data, and of interacting with the Navy's CMI system. The testing system is described, and the procedures for creating and administering the special diagnostic adaptive tests are discussed.

INTRODUCTION

Under a contract with the Office of Naval Research, a computerized testing system has been installed in the Basic Electricity and Electronics school at the San Diego Naval Training Center. The purpose is to extend to an actual training environment the theoretical work in cognitive error diagnosis being done by Kikumi Tatsuoka at the University of Illinois (Tatsuoka, 1985, 1986; Tatsuoka & Tatsuoka, 1986, in press). Using computerized tests, Tatsuoka's "Rule Space" model can diagnose the cognitive errors which underly a student's pattern of responses. The tests are adaptive, with the computer creating a path through the test items based on the accumulated responses of each student, then stopping the test when the student's errors are diagnosed.

This project is a classic example of the problems and decisions attendant on the application of a theoretical model to a "real world" situation. The goal of the project is to develop one or more cognitive error diagnostic tests for the Navy's Basic Electricity and Electronics curriculum using the Rule Space model, to administer these tests to pilot groups of Navy trainees, and to evaluate and report the results. This required the development of a sophisticated and flexible test administration system, an integrated system of software, a means for data collection and analysis, and test items designed specifically for this application. Furthermore, the testing system had to administer regular as well as adaptive tests, and to operate without disturbance to the training environment. A brief overview of adaptive testing is given, followed by a description of the testing system, progress to date, and some observations.

COMPUTERIZED ADAPTIVE TESTING

Adaptive tests are designed so as to administer to each examinee a set of test items matched to that individual's ability level. The computer searches through a stored bank of test items with known characteristics, selecting each time the next item in the bank which optimizes some statistical criterion function. As the examinee responds to each item, that information is used to update the item search information. When the desired criterion is reached, the test is ended (Green, 1983; Lord, 1980; Weiss, 1985). Figure 1 shows a general schema for an adaptive testing system.

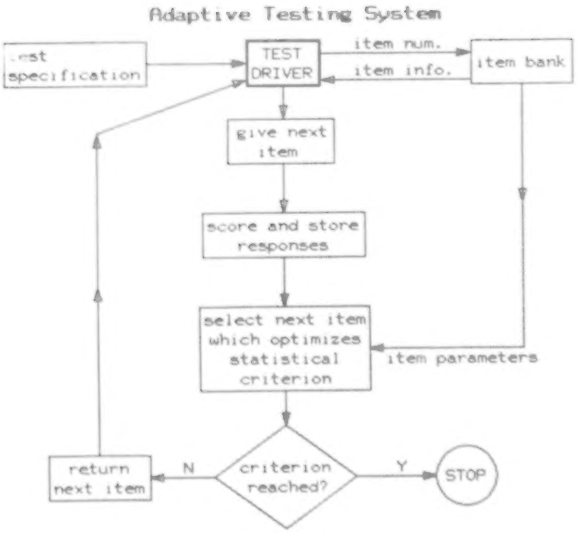


Figure 1. General schema for adaptive testing system.

In the case of the Rule Space, the computer plots a set of points corresponding to examinee responses for the test items selected. Once these points have clustered sufficiently, their centroid is compared with the centroids of other clusters known to represent various misconceptions or wrong rules of operation in the domain being tested. If the distance to one of these centroids is small enough to represent a high probability of a match, then the examinee's wrong rule is assumed to be diagnosed.

THE TEST ADMINISTRATION SYSTEM

System configuration. The design constraints for the proposed testing system required that it communicate transparently with the Navy's own CMI system, that it be capable of administering conventional as well as adaptive tests, and that it track and save test performance data. The system had to incorporate new testing strategies as they were developed, using high-level languages such as Pascal and FORTRAN. The microcomputer stations had to support acceptable color graphics, a local area network, and fast mathematical processing.

The test administration system, called MicroCAT, was designed and installed by Assessment Systems Corporation of St. Paul, Minnesota (Vale, 1985a). The hardware consists of 16 IBM-PC microcomputers linked by Ethernet to two server stations with hard disks, with these linked in turn via telephone modem to the Navy's CMI system. The software is written in Pascal, which can incorporate external procedures such as the FORTRAN programs for the Rule Space diagnostic testing.

Installation and testing. The MicroCAT system was installed and tested during the summer of 1984 (Vale, 1985a). Initial operation was restricted to simple linear administration of the Navy's existing tests, so as to separate the practical problems of test administration from experimental work with diagnostic testing procedures. Although the system has been operational since it was installed, ongoing revisions by the Navy of the instructional materials and tests delayed its full implementation until late March, 1985. Since then several thousand tests have been administered, and the system has continued performing without serious problems.

To create and install a test on the system, a bank of test items is developed using a graphic screen editor, plus a control file which lists the items to be administered and specifies the logical basis on which each successive item is to be chosen (Assessment Systems Corporation, 1984; Vale, 1985b). Most of the Navy's test items which have been installed on the system were developed by the project staff; however, some of the Navy personnel have become proficient enough to edit or modify test items, to re-compile and install tests, and to keep the testing system in operation.

As each trainee takes a test, the system stores a record which includes the item numbers, answer keys, student responses, and response times. These data are added to a binary file which is read periodically to diskettes and mailed to the project staff at the University of Illinois. Programs have been developed to convert the files from binary to text, to extract and score specific tests, to convert one test form to another, to create binary score vectors, to print summaries of the data, and to transfer the datasets to larger computers.

The diagnostic adaptive programs. The order in which the test items are administered can range from a simple sequence, as with the Navy's regular tests, to a decision computed by a program external to the test administration system and passed back to it. The MicroCAT system provides the means to attach custom scoring routines and other special procedures, so that the Rule Space programs can be incorporated directly into the testing system, and diagnostic adaptive tests using the Rule Space model can now be run from a floppy disc on an IBM-PC. Figure 2 is a generalized flow diagram of the programs.

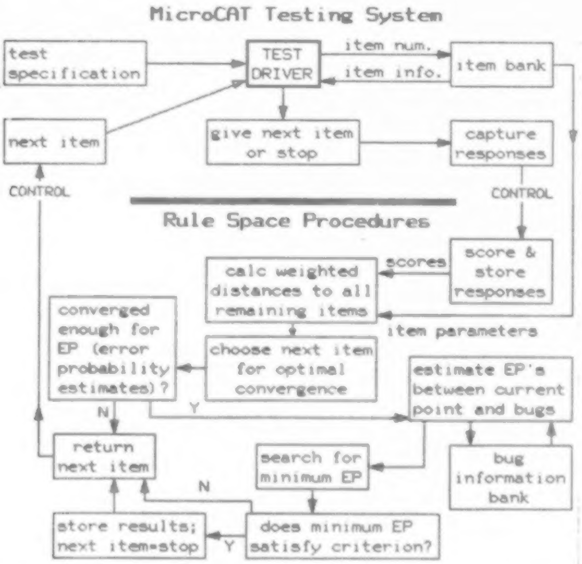

MicroCAT Testing System
test specification
TEST DRIVER
item num.
item info.
item bank
next item
give next item or stop
capture responses
CONTROL
CONTROL
Rule Space Procedures
score & store responses
scores
calc weighted distances to all remaining items
item parameters
converged enough for EP (error probability estimates)?
choose next item for optimal convergence
estimate EP's between current point and bugs
N
Y
return next item
search for minimum EP
bug information bank
N
store results; next item=stop
Y
does minimum EP satisfy criterion?

Figure 2. Linking Rule Space procedures to MicroCAT.

At the beginning of a test, item parameters for discrimination and difficulty which are stored in the item bank are read into an array in memory. As each test item is administered the examinee's response is captured, and program control is passed to the Rule Space procedures along with item response information. The examinee's response is scored as right or wrong, the weighted distances are calculated to those remaining items not yet administered, and the next item is selected for optimal convergence. When sufficient convergence is achieved so that the points have effectively begun to cluster, then the error probabilities are estimated between the point representing the current item and the centroids of a set of ellipses stored in a bug information bank. If the value of these probabilities satisfies a specified minimum criterion, next item is set for stop; if not, next item is not changed. Next item and program control are then returned to the test driver. In spite of the extensive processing required, lag time between administration of items is less than 1/2 second on the IBM-PC.

PROGRESS TO DATE

The mechanisms are now in place to administer diagnostic adaptive tests at the Navy's Basic Electricity and Electronics school. We are now in the process of creating test items appropriate for the Rule Space application. Once a complete diagnostic test has been developed, it will be administered to selected pilot groups of Navy trainees, and the results will be studied and reported.

In order to apply the Rule Space model to a test, estimated item parameters for discrimination and difficulty must be computed and detailed task analyses done for each test item. Item parameters are required by the Rule Space decision programs, but estimation of item parameters requires prior response data from a large number of subjects. This means that a test must first be developed, then administered to a significant population, and the item parameters estimated, before Rule Space can be applied to that same test. If item parameters for an existing test already are available the Rule Space programs can plot the response clusters; however interpreting the cognitive basis for the clusters requires a task analysis of the test items. Rule Space is not domain specific. A response cluster represents a statistical description of an examinee's path through a test, without reference to what is being tested. It is only through a task analysis of the domain being tested that the response clusters mapped by the Rule Space can be associated with specific cognitive errors; hence, the task analysis is necessary before Rule Space can be used for specific diagnoses. Since a task analysis is critical in any case, it is particularly important that the analysis also govern the design of new test items.

For the Navy's Basic Electricity and Electronics curriculum, test items must be constructed specifically for diagnostic adaptive testing, verified with a sample population, and item parameters estimated before the resulting test can be administered adaptively. The specific learning tasks addressed by the Navy's instruction, and the concepts, operations and errors represented by the Navy's tests currently are being analyzed in detail, in order to identify systematically the causes and results of various types of errors. Response data for the Navy's test items is being collected to aid in this task analysis, and also to estimate item parameters. So far, data for several thousand subjects has been collected, but these are divided among 30 different tests, and on most items a high percentage of the responses are correct, so that sufficient data for useful analysis has been slow in coming. However, item parameter estimates have been obtained for some of the tests, and analysis suggests that, with minor changes, selected items from the current tests can be used in diagnostic adaptive tests.

Performing a task analysis for any knowledge domain is intricate, tedious and time-consuming. Part of the problem stems from the complexity of logical inter-relationships, as represented by process networks. Ways are being investigated to use computers to speed up and simplify this aspect of the problem, leaving to humans the critical task of defining the relevant concepts and errors.

SOME OBSERVATIONS

The theoretical work by Kikumi Tatsuoka which has produced the Rule Space is sponsored by the Psychological Sciences Division of the Office of Naval Research. The purpose of this research is the investigation and development of psychological measurement models. Application of the Rule Space model in an actual training environment, as reported herein, is sponsored by ONR as a separate project, in order to determine if these techniques can eventually provide practical feedback for remediation in training.

From its inception, we have been careful to distinguish this application phase from the theoretical work. The working environments for research and applications are quite different. In matters such as time constraints, control of subjects and experiments, and tolerance of mistakes, the applications environment is far less forgiving than the experimental. Furthermore, a laboratory may develop one-of-a-kind programs or equipment, whereas the equipment and programs in a training facility must be standard, dependable and easy to use. For this project, the primary research has been done using the University of Illinois PLATO system for testing, data collection and for statistical computations and analysis. For the applied phase, the Rule Space programs are transferred to the IBM-PC, and integrated into the MicroCAT testing system, which has been independently tested in the training environment. Thus, the problems of setting up the testing system have been separated from those associated with the applications of Rule Space.

REFERENCES

Assessment Systems Corporation. (1984). *User's manual for the MicroCAT testing system* (Research Report ONR-85-1). St. Paul, MN: Author.

Green, B. F. (1983). Adaptive testing by computer. In R. B. Ekstrom (Ed.), *Measurement Technology and Individuality in Education: New Directions for Testing and Measurement*. San Francisco: Jossey-Bass.

Lord, F. M. (1980). *Applications of item response theory to practical testing problems*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Tatsuoka, K. K. (1986). Diagnosing cognitive errors: Statistical pattern classification based on item response theory. *Behaviormetrika*, *19*, 73-86.

Tatsuoka, K. K. (1985). A probabilistic model for diagnosing misconceptions by the pattern classification approach. *Journal of Educational Statistics*, *10*(1), 55-73.

Tatsuoka, M. M. & Tatsuoka, K. K. (in press). Bug distribution and statistical pattern classification. *Psychometrika*.

Tatsuoka, M. M. & Tatsuoka, K. K. (1986). Rule space. In Kotz & Johnson (Eds.), *Encyclopedia of Statistical Sciences*. New York: Wiley.

Vale, C. D. (1985a). Implementation of a microcomputer-based testing system in a military training environment (Research Report ONR-85-2). St. Paul, MN: Assessment Systems Corporation.

Vale, C. D. (1985b). MCATL: A language for authoring computerized adaptive tests (Research Report ONR-85-3). St. Paul, MN: Assessment Systems Corporation.

Weiss, D. J. (1985). Adaptive testing by computer. *Journal of Consulting and Clinical Psychology*, 53(6), 774-789.

Contact: John M. Eddins
252 Engineering Research Lab
103 S. Mathews Avenue
Urbana, IL 61801

217-333-6500

Using Technology to Measure the Effectiveness of Technology: Computers and Video as Data Collection Tools for Educational Research

Francis A. Harvey

Lehigh University

Abstract

In studying the effectiveness of various instructional technologies, educational researchers generally use pencil-and-paper instruments to measure learning and/or attitude changes and hand-recorded observation instruments to measure learner-instructional system or learner-learner interactions. Neither of these approaches takes full advantage of technology to aid the researcher in the collection and preliminary analysis of data. This paper presents several ways in which specific combinations of technologies, for example computers and videotaping, can be used to collect more complete and more accurate records of learning, attitudes, and interactions, and for easier and more reliable data analysis. Specific examples treated include the use of videotaped records, computer "dribble files," special-purpose computer programs which both evaluate and record experimental results, and combinations of the above approaches. The advantages and limitations of the various approaches employed are discussed.

Introduction

Two major types of instruments are generally employed by educational researchers to measure the effectiveness of technological-based instructional materials (instructional television, computer-based instruction, or interactive video). Pencil-and-paper instruments, either standardized tests or experimenter-developed instruments, are used to test both learning and attitudinal changes. In studies where interactions are of interest (either interaction between the learner and the technological-based instructional delivery system, or interaction between learners during the instruction), instruments which require trained observers to record their observations by hand on a structured observation instrument are generally used.

Each of these approaches has built-in limitations. When pencil-and-paper instruments are used to measure the effectiveness of technology-based instruction, the change in mode between technology and paper-based systems is a potential intervening variable. In observation studies, intensive training is usually required. Scoring the interactions during observation puts considerable pressure on the observer, and provides no means of checking the observations.

As an example of the limitations of the traditional approaches, consider the problem of investigating the relationship between learning to write computer procedures in Logo and children's higher-level thinking skills. Learning to develop microcomputer procedures in Logo is a complex task, involving a wide range of cognitive processes. A key component of the overall process is the amount of time a learner spends analyzing screen information and formulating his or her response. In addition, such learning is influenced by the user's attitudes toward the computer and (in small group situations) by the learner's attitudes toward other learners in the group. In order to study the actual learning strategies a group of learners use in planning, writing, and debugging Logo procedures, a complete, minute-by-minute record of what the learners entered on the keyboard, what information the computer displayed, and what the learners said to each other is needed. Gathering such a record is a formidable task, and traditional instruments are clearly not adequate for such a task.

This paper will argue that educational researchers have not fully utilized the potential of various instructional technologies as tools for data collection and analysis. Specific combinations of technologies, for example computers and videotaping, can be used to collect more complete and more accurate records of learning, attitudes, and interactions. Technology can also be used for many tasks involved in the preparation of data for formal analysis, eliminating the effort and potential for introducing error involved in hand scoring and hand coding data.

This paper will present and discuss several ways in which specific combinations of technologies, for example computers and videotaping, can be used to collect more complete and more accurate records of learning, attitudes, and interactions, and for easier and more reliable data analysis. The specific examples treated include 1) the use of videotaped records, either alone or in combination with computer-generated records; 2) computer-based "dribble files," which can generate a running, time-coded record of all of the instructions entered into the computer; 3) special-purpose computer programs, embedded in the instructional program, which evaluate both the learning process and learning outcomes and record the results of both on disk for later analysis; and 4) combinations of the three approaches. The advantages and limitations of the various approaches employed will be discussed.

Videotape as a data collection tool

Interactions between the learner and the instructional system, or interactions among several learners in a group situation, are complex and difficult to measure. By videotaping experimental sessions, experimenters are able to generate a complete and accurate record of what the subjects did and said during the experimental session.

Videotaped records have been used to measure learners' reactions to both computer-based instructional materials and interactive videodisc-based materials. Bigum, C.J., and Gilding, A. (1985) used a video monitoring technique to investigate children's reactions to computer-based science instructional programs. They conducted their experiment in a television studio, and used two video cameras and a video mixer to obtain a videotape record of the children's actions with the computer display in a small inset in one corner of the videotape image. The videotape thus provided a complete, time-coordinated record of both the learners' behavior and the treatment to which they were responding. Kirchner, Martyn, and Johnson (1983) used videotape to evaluate children's attitudes toward learning from an interactive videodisc-based unit on the heart and circulatory system.

Videotaping provides data on both the spoken responses and body language of the subjects. These provide information about subjects' attitudes toward both the material being presented and the environment in which it is being presented (which includes other subjects, if present, as well as the hardware/software components of the instructional system).

The videotaped record can be analyzed at a later time, away from the experimental setting, at the convenience of the experimenter. Reliability of raters can be established by having more than one rater score the same videotape. The videotapes serve as a permanent record of the experiment, and are available for further analysis at a later date if desired. Experimental videotapes can also be edited into a demonstration videotape which can be used to explain the project to funding agencies or other educational researchers. Finally, videotapes can be used as protocols for the training of observers for additional studies, and for analysis and refinement of the experimental procedures.

Dribble files

Dribble files are created by special computer programs which are loaded into (or are built into) the computer's memory and operate at the same time as the programs which are providing the treatment in a computer-based educational research experiment. Alternately, the programs which generate the dribble file can be a set of procedures which are included as part of the program providing the treatment. The name comes from the term used for such files in the version of Logo developed by Logo Computer Systems, Inc. for IBM and MS-DOS microcomputers.

Dribble file procedures intercept all instructions entered as keyboard input and copy that input to a special area of memory before interpreting and executing the instructions. Dribble files are transparent to the user; the user is unaware of their operation. Periodically, a record of all of the input is copied from the computer's memory to a data file on the microcomputer disk. Dribble files can be examined after the experimental session, and all of the subject's input can be analyzed.

Dribble files provide the opportunity to observe and analyze not only the final response of the subject to a specific prompt, but all of the intermediate responses generated as well. Thus the "incorrect" answers, "false starts," and "changes of mind" are all available for analysis. Such a record is more valuable for analyzing the process of learning than a summary of the subject's input at specific points in a program.

As mentioned above, the capacity to produce dribble files is built in to IBM Logo. Machine language programs which intercept all keyboard input and save it before processing it can be developed for other applications. At Lehigh, we have used both Logo dribble files and a set of procedures named RECORD, developed by graduate student Peter Geyer for the Apple II microcomputer series, to generate dribble files.

Special purpose computer programs

Dribble files have considerable utility for educational research. They are limited, however, in that they do not provide a time-coded record of the interaction. The amount of time spent before responding is of interest in many studies, e.g., in analyzing the relationship between the response time of an interactive videotape or videodisc system and subjects' interest and performance.

Another example in which time for response is of interest is in the study of learners' preference for various means of displaying information (e.g., tables, bar charts, or line graphs) in interpeting and understanding that information. A record of which display type the subject selected for specific questions (generated and presented by the computer program), the sequence in which various display types were selected, how long the subject examined each display type, and how accurate the subjects' responses were would provide valuable data on the subject's style of information processing.

To study time taken for subject responses, special purpose microcomputer programs can be developed which constantly check the keyboard for a response and record the time taken for such a response. These programs can use either the microcomputer's built-in clock, if the microcomputer is so equipped, or a specially written machine language timing program.

Special-purpose data collection programs can also be used to record the results of post-treatment measures which are administered and scored as part of the microcomputer program which provides the treatment. The experimenter, at the end of each experimental session, has a permanent record on a microcomputer disk of the subject, date, questions asked, time for response (if desired), raw responses to specific criterion measures, and summary data on these responses. The tedium of converting evidence of the learner's responses from pencil and paper or some other non-computer form back into a form which the computer can analyze, and the attendant risk of introducing errors in transcribing or converting, can be eliminated if the data collection is incorporated into the treatment itself.

Combinations of videotape and computer-generated records

Each of the applications described above can be effective when used by itself. An even more effective application, however, may be to use combinations of technologies as a data gathering system.

A project currently under way at Lehigh attempts, as a first step in developing suitable tools for measuring the relationship between programming in Logo and higher-level thinking skills, to use various combinations of technologies to gather a continuous, complete record of the learner-computer interaction during a Logo programming session. It is hoped that analysis of such records will provide a foundation for the development of specific instruments which can be used to measure higher level learning outcomes.

A key component of the project is the use of videotape equipment, in combination with other data-collection tools, to collect a complete record of children's involvement with and reaction to Logo. Video equipment will be set up so that a continuous record will be gathered of all keyboard inputs by the children, the resulting changes in the computer's video display, and the children's verbal and motor behavior in response to the computer.

The project, which is sponsored by a Lehigh University research grant, will run from February to September, 1986. Various hardware configurations and environments will be designed and pilot-tested to evaluate children's learning from several different Logo microworlds. The strengths and limitations of each hardware configuration and setting will be evaluated and analyzed. Particular attention will be paid to the relative cost and ease of use of each configuration.

The use of three types of technology will be investigated:

-- videotape, using existing videotape equipment and facilities owned by the Lehigh University College of Education, the Educational Technology Program, the Summer Session office, or the Lehigh Media Center;

-- microcomputer "dribble files," which write a complete record of all keyboard entries to a disk file;

-- audio tape.

Various combinations of these technologies will be evaluated in several different settings. Present plans call for pilot testing the proposed procedures 1) in a standard television studio, equipped with multi-camera capability and a video switcher, and 2) with portable video equipment in field settings which would be more like the environments in which data on Logo programming is typically collected.

An additional feature of the project will be investigation of the use of other inexpensive equipment, for example, a sweep-second or digital timer the image of which could be superimposed on the video image of the learners.

Summary

The potential of computers, video, interactive video, and other technologies for effective instruction has long been recognized by educators and intensively studied by educational researchers. It is hoped that the analysis and systematic application of ways in which the same or similar technologies can be used to support and enhance the data-gathering and data-analysis components of educational research will bring about similar gains in the effectiveness and efficiency of the educational research process.

References

Bigum, C.J., and Gilding, A. (1985). A video monitoring technique for investigating computer-based learning programs. Computers in Education, 9, 95-99.

Kirchner, G., Martyn, D., and Johnson, C. (1983). Simon Fraser University videodisc project: Part two: Field testing an experimental videodisc with elementary school children. Videodisc, videotext, 3, 45-58.

Contact: Francis A. Harvey
College of Education
Educational Technology Center
Lehigh University
301 Broadway, Third Floor
Bethlehem, PA 18015-1598

215-758-3231

ADCIS Proceedings, November 1986

THE EFFECTIVENESS OF DIFFERENT RESPONDING APPROACHES IN MULTIPLE-CHOICE OF CAI DESIGN

Hong, Jon-Chao

National Taiwan Normal University

ABSTRACT

The implementation of the computer in education has become an essential tool in instructional settings. Differing from the coventional way, computers used in multiple-choice design allow students to employ many approaches, Furthermore, it is assumed that different ways to aquire the learning materials will result in different outcomes of learning.

In addition to the conventional approach, this study examines three new approaches: highlighting, underlining, deleting to act upon those characters appearing on the line of the screen to find the most effective approach. The result of this study indicates that the "highlighting" approach is the most effective.

INTRODUCTION

The computer has become a very important tool in the instructional settings. It has led many educators to develop quite a variety of coursewares to assist teachers to conduct their instruction in various fields. In the past, the use of the computer in teaching has not been as effective as expected, because it has not taken into account the human factor and cognitive development in the designing of computer-assisted instruction (CAI) (Jay, 1983).

As the computer faclitates in teaching, CAI screen design can be constructed in many formats. Different screen designs can result in various cognitive responses. For this reason, this study focuses on how the responding approaches in CAI multiple-choice design affects the retention of students' learning. In addition to the traditional one, this study uses three new responding approaches: highlighting, underlining, and deleting. All of them test the effectiveness of students' learning by employing their respective function on those characters on the line of the target words.

Purposes of this Study

The purpose of this study is to answer the following questions:

Is there any significant difference in the effectiveness of the approaches on the basis of:

(1) educational levels (3rd and 8th grades),

(2) stimulus materials (easy or difficult to the samples).

Hypotheses

In order to test the effectiveness of different approaches in the CAI multiple-choice responses, the specific null-hypotheses of this study are constructed as follows:

Ho1: There is no significant difference in the effectiveness of learning among the four approaches in the CAI multiple-choice responses.

Ho2: There is no significant difference in the retention of learning resulting from the four different approaches in the CAI multiple-choice responses (based on the time span between the two post-tests).

Ho3: There is no significant difference in the retention of learning based on the educational levels resulting from the four approaches in the CAI multiple-choice responses.

Ho4: There is no significant difference in the retention of learning based on the stimulus materials resulting from the four approaches in the CAI multiple-choice responses.

SURVEY OF THE LITERATURE

The effectiveness of learning depends on the variety of computer functions. Its effectiveness can usually be assessed by the retention rate of learning. Therefore, the literature survey mostly is concerned about the relationship between retention of learning and the application of CAI design.

Morgna and Richardson (1974) researched the CAI used in math. education on 7th grade students, and found that the immediate effect of CAI was better than traditional instruction. Yet, after three months, the retention rate of the CAI test group was the same as that of the control group.

Karon (1975) conducted a study among medical school students and found that the retention rate of the CAI group was closer to that of control group as the time went on. Jackson (1976) did a similar research on the students in the educational administration department and found the same result as Karon did.

From these studies, it may be concluded that the retention rate in the CAI significantly increases immediately following the experiments.

Moreover, Clark (1982) addressed that the human information process could be thought of "perceptual link" and "meaning link." If one wanted to increase the quality of stimuli, the perceptual link should be strengthened. "In other words, the higher the quality of stimuli, the longer the retention rate will be extended. Whereas Seymour (1979) pointed out that using pictorial interface could increase long-term memory, which means that the perceptual link can be strengthened by pictorial interface through mind functions.

Therefore, this study emphasizes the variety of pictorial interface on the effectiveness of the CAI multiple-choice design. The pictorial interfaces designed in this study were: highlighting, underlining, and deleting. All of them acted upon the target characters appearing on the screen which displayed questions presented by the CAI multiple-choice.

RESEARCH METHOD

The Sample of this Study

There were 110 third grade students from two elementary schools and 133 eight grade students from two junior high schools, who participated in this experiment. They were divided into four groups:

(1) The controlled group. As the multiple-choice questions appeared on the screen, the participants of this group responded by choosing a number and them pressed the "RETURN" key to answer the questions. This group was named: Controlled Group.

(2) The highlighting group. As the participants of this group responded to the multiple-choice questions, they used either " (" or ") " key to move the "cursor" back to the beginning position of that statement. Then they pressed "spcae bar" to highlight all the characters in that statement. Finally, they pressed "RETURN" to complete the answering approach. This was labelled: Experimental Group I.

(3) The underlining group. The process of the answering approach was similar to the "highlighting" group, but they pressed "space bar" to underline all the characters in the given statement. This group was labelled: Experimental Group II.

(4) The deleting group. The process of the answering approach was similar to the latter two experimental groups, but they pressed "space bar" to delete those characters in the given statement. This meant that those characters on the line of questions would disappeared from the screen when the participants pressed the "space bar." This group was named: Experimental Group III.

Stimulus Materials

The stimulus materials focused on developing a "practice" of CAI model on Apple II computer to allow the samples of this study to complete 40 questions which were generated from the software. The participants of this study had to answer each question until they answered correctly.

Those 40 questions were constructed as follows: (1) gear transition, clock reading, Chinese word retrieval, and Chinese phonetics for third grade students to answer; (2) sementic reasoning, puzzle board transformation, solar system, and Taiwaness geography for eigth grade students to anser.

Each subject consisted of 10 different questions, and five out of ten were selected for post-tests, which altogether made 20 questions. After, the participants of this study completed these stimulus materials on the computer, they received two post-tests. The first post-test was given the day after the preliminary experiment. The second post-test was given ten days later.

FINDINGS

The score of two post-tests were analyzed by 3-way ANOVA with unweighted-means solution and Schefee test. Briefly, the results of this study can be explained as follows:

(1) How the four responding approaches affected the retention rate of third grade students.

According to the data analysis, the result of this part of the study indicated that the "underlining" approach was worse than the "controlled and "highlighting" approaches ($p < .001$), but not worse than the "deleting" approach ($p > .01$). The "highlighting" approach was more effective than either "underlining" or "deleting" ($p < .001$). Whereas the effectiveness of the conventional approach was the same as that of "deleting." The result of this part of the study can be concluded that the "highlighting" approach seems to be the most effective among the four approaches.

(2) The effectiveness of the four responding approaches on the eigth grade students.

Based on the Schefee test, the result of this part of the study proved that the effectiveness of the "highlighting" approach was better than those of "controlled" and "underlining" ($p < .001$) but not better than the "deleting" approach ($p > .05$). Whereas, the effectiveness of the "underlining" approach was worse than either "highlighting" and "deleting" ($p < .001$), but not worse than that of "controlled" ($p > .05$) in the samples of eigth grade students.

(3) The effectiveness of the responding approaches on the difference between easy and difficult stimulus materials.

Based on the data analysis, there was no evidence to address that the type of stimulus materials in the responding approaches influenced the retention of learning ($p > .05$).

From the evidence of this study, the retention rate of the "highlighting" approach significantly decreased from the first post-test to the second post-test ($p < .05$), but it is not decreased in the other three approaches. This result seemed to support Karon's and Jackson's finding.

CONCLUSION

Differing from the traditional instruction, the computer can be designed in various ways to display its features. Based on this function, this study employs the responding approaches in the four different ways to allow students to answer the multiple-choice questions. The respective functions of the four approaches are discussed in the "method" section of this study.

At any rate, the evidence shown from this study can be explained as follows:

(1) The effectiveness of the responding approach is not influenced by the stimulus materials which are classified as "easy" and "difficult" type question.

(2) The effectiveness of the "highlighting" approach is the most significant in the retention of learning.

(3) Although the "highlighting" approach is the most effectiveness among the four responding approaches in the retention of learning, the retention rate influenced by the "highlighting" approach decreases with time.

The result of this study implies that using the "highlighting" approach in designing is not only the most effective method in the CAI multiple-choice, but also in other screen texts.

REFERENCE

Clark, R. E. (1982) Instructional methods: operational definitions and criterial attributes. Paper Presented at the American Educational Research Association, New York.

Jackson, J. A. (1976) A comparative study of the effectiveness of programmed instruction and computer-based instruction. Dissertation Abstracts International, 37, 6335-A.

Jay, T. B. (1983, Jan.) The cognitive approach to computer courseware design and evaluation. Educatiuonal Technology.

Karon, L. (1975) An experimental study comparing computer-assisted instruction with the traditional lecture method in an introduction learning disabilities course give to medical students. Dissertation Abstracts International, 36, 4226-A.

Morgan, C. Richardson, W. (1974) Computer-assisted instruction program, Rockville, Maryland: Board of Education of Mortogmery.

Seymour, P. H. K. (1979) Human visual cognition, New York: ST. Martini.

Contact: Jon-Chao Hong
Dept. of Industrial Education
National Taiwan Normal University
Ho-Ping E. Road
Taipei, R.O.C.

02-321-9742

STRATEGIES FOR PERFORMANCE AND LEARNING WITH AN INTERACTIVE COMPUTER SYSTEM

Elaine Lewis
Laboratory for Instructional Technology, Boston University

Victor Shtern
MET Computer Science, Boston University

Abstract

This paper proposes a model for performance and learning based on systematic observation of seventy-two subjects using an interactive computer program. Aimed at defining basic cognitive strategies for educational software, this technique could lead to more accurate prediction of performance from computer simulation programs as well as increase general understanding of how people learn to use interactive computer systems.

Introduction

As the cost of hardware and memory decreases, and the number of general purpose and specialized mini- and microcomputer systems grow, an ever increasing number of people use computers in the course of their daily life. Educational applications add to this use. Today the number of non-programmer and novice users greatly exceeds that of professional programmers, hence knowledge about the behavior of beginning users has become crucial for system effectiveness and reliablility (Shneiderman, 1980).

For over a decade, system developers have realized that greater knowledge of user psychology is necessary, but little basic research has addressed this area (Miller and Thomas, 1977). There is also a great need for empirically-based criteria which can aid the design of programs that successfully match the capabilities and limitations of their intended users. These criteria need to be shaped by knowledge of how people interact with computer systems. Systematic description of this interaction process is a prerequisite to address more fully defined cognitive models which relate to more complex behaviors and learning (National Research Council, 1984).

This study proposes a scheme for categorizing computer users according to two dimensions of the cognitive strategy which they employ while acquiring and understanding information. The underlying approach assumes that computer users are heterogeneous in terms of their cognitive style, and that this style can be discovered through careful observation of their interactions with computer programs.

Cognitive Strategies for Acquiring Information

Certain fields in applied cognitive psychology, such as learning theory, have successfully predicted performance and learning based on an individual's cognitive strategy for acquiring and using information (Kolb, 1976, 1981; Witkin and Goodenough, 1981). Central to this literature is David Kolb's Learning Style Inventory (LSI), published in 1976, which defines four categories of learning style.

These categories arise from the intersection of two cognitive dimensions: "concrete/abstract" and "active/reflective." Both dimensions relate to an individual's most successful methods and modes for assimilating information. The first dimension, based on the work of cognitive theorists (Bruner, 1966; Harvey, Hunt and Scroeder, 1961), represents peoples' relative success at understanding ideas and events through concrete experience as opposed to abstract conceptualization. An "abstract" person might understand a phenomenon or process most clearly through a conceptual model or theory of the process, while a "concrete" person needs actual life experience of that process to understand it fully.

The second dimension of Kolb's typology, drawn from the work of Piaget (1951) and others, represents an individual's preferred method for acquiring information, from active experimentation at one extreme to reflective observation at the other. In other words, people accept or reject ideas through different methods: they may actively test their hypotheses, or they may reflectively interpret previously gathered data. Although this dichotomy is usually discussed with regard to age-dependent stages in child development, others have observed these differences in mature adults (Even, 1982).

The combination of these two dimensions generates four possible types of adult learners: "divergers" who prefer concrete experience and reflective observation; "assimilators" who prefer abstract information and reflective observation; "convergers" who prefer abstract information and active experimentation; and "accommodators" who prefer concrete experience and active experimentation. These categories form the core of a learning model which implies that ability to learn and communicate at school or work is related to successfully matching cognitive style to the learning or work environment.

A few researchers have considered individual cognitive styles in terms of how people learn to use computers (Magers, 1982), and how they solve problems with information systems (Grochow, 1973). However, none of these studies consider how typologies of cognitive style should relate to the design of computer software. The few studies which have tried to develop empirically-based guidelines for software design usually characterize computer users as a homogeneous group.

In one experiment examining people's use of interactive communication programs like electronic mail systems, Hiltz and Kerr (1986) found that most people relied upon a trial-and-error method rather than resorting to manuals, human experts or on-line help facilities. (Paradoxically, the trial method is one of the least efficient in terms of learning speed.) Their finding led to the suggestion that software for interactive communication systems should be designed to support trial-and-error learning. Unfortunately, their experiment did not explore how variations in cognitive strategies might affect their subjects' choice of learning mode. Sensitivity to the variations in cognitive strategy would enable the development of more specific, and therefore more effective, recommendations for software design.

Cognitive Strategies for Computer Users

The learning theories reviewed in the previous section suggest that cognitive style falls into one of four categories. Before this typology can be adapted to describe computer users, the cognitive dimensions must be operationalized through characteristics of computer software and relevant interaction variables. The remainder of this paper will describe an initial experiment designed to examine one of these dimensions of cognitive style: active/reflective.

In contrast to previous research which often relies on self-reported, survey methods for categorizing individuals (Kolb, 1976; Nunney, 1978), this study employs observational techniques to establish the dichotomy. The active/reflective dimension is defined by observing how a user solves a series of tasks presented in a computer program. Each subject has the option of referring to help screens and references. Whether an individual approaches this process in a systematic fashion, or attempts to solve the tasks through trial-and-error experimentation will determine whether that person will be categorized as having an "active" or "reflective" cognitive style.

These style categories are hypothesized to predict differences in subjects' performance and learning with a computer program. Ultimately, cognitive strategies will be correlated with specific program characteristics, such as level and type of interactivity. By determining an individuals' or class of individuals' cognitive strategies, software designers will be able to create more effective programs which facilitate acquiring and using information.

Stimulus Program and Procedure

The interactive computer program used to determine cognitive category and evaluate performance took about thirty minutes for subjects to complete. The program presented a series of nine location tasks which could be performed by trial-and-error or through systematic reference to help screens and descriptive resources.

The program began with a brief introductory segment which set the scenario. Subjects were asked to "visit" Washington D.C. vicariously via a series of computer-generated maps of the city. During the visit, they were expected to learn specific details about nine different tourist sites and their locations on the maps. They were informed that later in the program they would be asked to visit each site by touching the appropriate map to indicate its correct location.

After detailed text and graphics described each of the sites, subjects were confronted with nine scavenger-hunt style missions which required them to identify sequentially the nine different locations. Each location had a unique associated task, and task presentation order was randomly assigned in an attempt to avoid confounding the order of tasks with their relative difficulty.

Once the tasks began, subjects were confronted with a basic map which divided the city into seven geographic sections. To travel to one of these sections, the subject touched its representation on the basic map; the screen cleared; and a close-up map of that area was displayed. The subjects then had to identify the location of the site described in that specific task.

Individual subjects were asked to view the stimulus program using a DEC PRO350 workstation with color graphic capabilities and a touch sensitive screen. The experiment was conducted in Boston University's Laboratory for Instructional Technology. In each case a research assistant was present to ensure consistency in the experimental procedure.

After a brief introduction to the system which included an explanation of the experiment, each subject was asked to view the program and follow all instructions. The system automatically recorded elapsed time for the entire program, and for each task, along with the number of incorrect location attempts and requests for help clues and information. After finishing the program, subjects were asked to complete an exit questionnaire measuring their satisfaction with the program along with demographic information and a measure of computer experience.

__Evidence for Active/Reflective Categories__

Two variables were used to create cognitive style categories, the number of requests for information and the number of incorrect responses. Subjects were divided into dichotomous groups based on the median of each variable. Those with more than seven requests for information were placed in the "many references" category, while those with more than twenty-five incorrect responses were categorized as having "many incorrect". Table 1 shows the distribution of subjects according to these variable categories.

Both variables reflect behaviors that should depend on whether people employ an active or a reflective cognitive style. Subjects who systematically refer to help screens and other information demonstrate an information-seeking style based on a reflective orientation. Those who adopt a trial-and-error approach can likewise be categorized as having an active style. The degree to which subjects engage in trial-and-error behavior is indexed by the number of wrong answers they give in response to the tasks. In order to be categorized as "active," subjects had to request few references and show many incorrect responses. Those with many references and few incorrect responses were considered "reflective." In Table 1, subjects with an active style appear in the upper right-hand cell, while those with a reflective style appear in the lower left-hand cell.

	few incorrect	many incorrect	
few reference	n=14 19.4%	n=24 33.3%	n=38 52.8%
many reference	n=22 30.6%	n=12 16.7%	n=34 47.2%
	n=36 50.0%	n=36 50.0%	n=72 100.0%

Table 1: Distribution of Subjects According to References and Incorrect Responses

Nearly two-thirds of the seventy-two subjects fall into one of the cognitive style categories. A corrected chi-square, significant at well beyond conventional levels, indicates that this relationship is probably not a chance occurrence (χ^2 =4.52, df=1, p=0.03). Another statistic, gamma, suggests that number of references and number wrong vary inversely and that their relationship is relatively strong (gamma=-0.52). Indeed, these findings may be compelling evidence that most people exhibit either an active or a reflective style when interacting with computer programs of this type.

Subjects who had few references and few incorrect responses, along with those with many references and many incorrect responses, cannot be placed in a category.

It is likely that their scores on these measures should be attributed to other factors like anxiety, memory, or reasoning ability. While these are important variables to pursue, they are beyond the scope of this initial study.

The relationship between use of references and error rates depends upon a subject's level of experience with computers (χ^2=15.34, df=6, p=0.01). Three categories for computer use -- never, moderate (under ten hours per week), and high (over ten hours per week) -- were compared to the style variable categories. This analysis produced some surprising results. Almost half of the people who said that they never used computers appeared in the reflective category (43.8%), while more than half of the moderate users were classified as active (53.8%). Heavy users were evenly distributed between active and reflective categories (30% in each).

These results suggest that peoples' cognitive strategies for using computers may evolve as they achieve certain levels of use. In this experiment, novices tended to be reflective, while moderate users were more often active. Perhaps only when people perceive the computer as a common tool, as may be the case with heavy users, will evidence of distinct and evenly distributed style categories emerge.

More support for the notion that experience factors affect active and reflective styles appears through examining characteristics of subjects who don't fall within one of these categories. Of the people who made many references and gave many incorrect responses, half never use a computer in an average week. Over half of the people in the few references and few wrong category (57.1%) were heavy computer users.

Implications for Future Research

This experiment has been successful in achieving its objectives. The observational technique used to establish computer users' active or reflective cognitive style yields discrete categories of users. Findings also suggest other independent variables, like level of computer experience, as factors influential for determining cognitive strategy.

Future studies must focus on ways to relate the active/reflective cognitive dimension more directly to computer software characteristics. This same line of inquiry should strive also to encompass the concrete/abstract dimension in order to complete the proposed typology for cognitive strategies.

REFERENCES

BRUNER, Jerome S., Toward a Theory of Instruction, New York, Norton, 1966.

EVEN, M.J., "Adapting Cognitive Style Theory in Practice", Lifelong Learning: The Adult Years, 1982, vol. 5, no. 5, pp. 14-17.

GROCHOW, J., "Cognitve Style as a Factor in the Design of Interactive Decision Support Systems", unpublished Ph.D. Thesis, Cambridge, MA, Massaschusetts Institute of Technology, 1973.

HARVEY, O.J., David Hunt and Harold Schroder, Conceptual Systems of Personality Organization, New York, John Wiley, 1961.

HILTZ, Starr Roxanne and Elaine B. Kerr, "Learning Modes and Subsequent Use of Computer-Mediated Communication Systems", Human Factors in Computing Systems, CHI'86 Conference Proceedings, Association for Computing Machinery, 1986, pp. 149-155.

KOLB, D.A., Learning Style Inventory Technical Manual, Boston, McBer, 1976.

MAGERS, C.S., "The Relationship Between Learning Style and Successful Computer Use by First Time Computer Users", unpublished Master's Thesis, Worcester, MA, Clark University Graduate School of Management, 1982.

MILLER, L.A. and J.C. Thomas, "Behavioral Issues in the Use of Interactive Systems", International Journal of Man-Machine Studies, vol. 9, 1977, pp. 509-536.

NATIONAL RESEARCH COUNCIL, Committee on Human Factors, Commission on Behavioral and Social Sciences and Education, "Research Needs on the Interaction Between Information Systems and their Users: Report of a Workshop", 1984.

NUNNEY, D.N., "Cognitive Style Mapping", Training and Development Journal, 1978, vol. 32, pp. 50-57.

PIAGET, Jean, Play, Dreams, and Imitation in Childhood, New York, Norton, 1951.

SHNEIDERMAN, Ben, Software Psychology: Human Factors in Computer and Information Systems, Boston, Little, Brown and Company, 1980.

WITKIN, H.A. and D.R. Goodenough, Cognitive Styles: Essence and Origins, New York, International Universities Press, 1981.

Comparison of Foreign Language Learning with Computer-Assisted and Non Computer-Assisted Study

Mary Ann Lyman

University of Idaho

Introduction

In response to national mandates for communicative competence in foreign languages, current foreign language methods stress the oral skills of listening and speaking. Heavy classtime focus on oral skills requires some measure of out-of-class preparation and compensation in the other areas of communication, reading and writing.

Textbooks are usually accompanied by workbooks to provide the student with more opportunity to develop writing skills. The workbooks are often "self-correcting," requiring the student to look up answers in the back and to change incorrect responses. Explanations are seldom provided in the key, so students may not learn from correcting mistakes alone. In large beginning language sections, checking that students have caught all errors may be tedious and difficult for the instructor. Lalande (1984) demonstrated that corrected written work is discouraging to the beginning student, who is likely to make many errors.

Purpose of the Study

The computerized "electronic workbook" is sometimes cited as a remedy to the difficulty of correcting students' written errors (Suppes, 1965; Hope, 1985). Sanders and Kennner (1983) also believe that computer-assisted language instruction (CALI) offers much promise in individualizing instruction according to learning styles.

Oral drill and practice have been said to enhance language acquisition (Stansfield & Horner, 1981); however, students learn at different rates and have different learning styles (Corbett & Smith, 1984; Reinert, 1976). The oral pattern drills of the new methods may not be as useful for students whose learning preference mode is WRITTEN WORD(WW). Students citing a preference for this learning style as compared to those indicating a preference for listening were less successful in a beginning Spanish class, according to the criterion measures used to ivaluate reading proficiency in an experimental setting described by Corbett and Smith in 1978. Language educators feel that the comprehensive performance studies are needed to illustrate the effectiveness of computers in foreign language study (Taylor, 1981). The purpose of this study, therefore, was to create teacher-authored courseware to accompany a beginning language course, make the programs available to students for review purposes, and determine whether at the end of the course, differences in writing and listening proficiency existed between those using the courseware and those using the traditional workbook methods of review. To correct for initial differences among students, measures were used in the data analysis concerning general scholastic aptitude and prior experience in the language.

The Hypotheses

Data were collected for students in French 101 (beginning French at the university level) as to prior experience in the language, age, sex, and general academic ability. Following the course, the Modern Language Listening Comprehension test and a common final examination for French 101 were administered. The common final examination was constructed with the collaboration of French instructors at the University of Idaho. Data were gathered for general academic ability as measured by either the Scholastic Aptitude Test (SAT) or the ACT Battery of the American College Testing Program (ACT). Analysis of covariance was used to test the following hypotheses:

> There is no significant difference in writing proficiency between groups of university level beginning French students. One group has chosen to review for chapter tests using computer assisted instructional programs and the other used traditional workbook study methods. Writing proficiency is determined by performance on the written final comprehensive exam (CE), constructed at the Novice High level of the ACTFL/ETS Proficiency Guidelines.

$$H : \bar{M}\ CEgp1 = \bar{M}\ CEgp2$$

$$H : \bar{M}\ CEgp1 <> \bar{M}\ CEgp2$$

> There is no significant difference in listening proficiency between groups of university level beginning French students. One group has chosen to review for chapter tests using computer assisted instructional programs and the other used traditional workbook study methods. Listening proficiency is determined by performance on the Modern Language Association Listening Comprehension exam.

$$H : \bar{M}\ MLAgp1 = \bar{M}\ MLAgp2$$

$$H : \bar{M}\ MLAgp1 <> \bar{M}\ MLAgp2$$

Subjects

The subjects selected for this study were all university students ranging from eighteen to thirty years old. All were regularly enrolled in French 101, a non-required elementary course, during the fall semester of 1984. Four sections of French 101 were offered. Two were non-randomly selected for participation in the study.

Procedures and Instrumentation

Courseware had been written in SUPERPILOT the previous summer to supplement French 101. All four sections of French 101 in the Fall semester of 1984 were invited to participate in the software program. Students were invited to attend introductory software sessions and were then asked if they would like to help "test" and "debug" the programs. They were told that no extra credit could be given, but that they might find the programs helpful as a review tool.

The two sections selected to be included in the study were taught by the same instructor and thus held the classroom variable constant. Students choosing software to review filled out a questionnaire evaluating each diskette. A feature of Superpilot also recorded the student's name on the diskette as a double-check.

Statistical Methodology

The final examination scores were recorded in percentage of items correct. Those students who did not have SAT or ACT scores were discarded from the analysis, and, where SAT scores were not available, ACT scores were converted to SAT scores using a conversion chart from the Student Counseling Center of the University of Idaho. Those students who did not take the MLS Listening Comprehension Test were also discarded from the study

The data were analysed using the SPSSx (Statistical Package for the Social Sciences). After a test for homogeneity of variance, analysis of covariance with SAT scores and experience scores (recorded in units equivalent to high school semesters) compared the performance of the two treatment groups. The use of the analysis of covariance helped control statistically for initial differences among students which might have been present and which might confound differences between the computer and the non-computer group.

Results of the Analysis

The student data are presented by treatment group, and together with the descriptive statistics are reported in Table I.

Table II shows the results of the analysis of variance with the effects of prior experience and scholastic ability as covariates and the dependent variable specified as the score obtained on the examination in French.

The first hypothesis, that no significant difference in writing proficiency existed between treatment groups, was rejected at the .05 level. Group participation was found to significantly influence performance in written French as determined by the final examination scores for French 101.

Considering the same covariates, Table III outlines the effect of group participation on the Modern Languages Listening Comprehension test. The second hypothesis, that no significant differences exist in listening comprehension ability between treatment groups, was not rejected at the .05 level of significance. Prior experience appeared to be the most important factor determining success for those students taking the MLA listening test, but not significantly so. Group participation had little or no effect on the outcome of the MLA tests.

Summary and Conclusion

Although this study found significantly higher scores in writing proficiency for those using the computer-assisted study method of review, there was no concurrent effect on listening comprehension. It is to be noted, however, that the sample was altered to remove those cases for whom data were missing.

As the proficiency movement continues in emphasis, we will look for means of maximizing the productivity of out-of-class time student preparation. Computer-assisted instruction may offer one method to teach the "building blocks" of grammatical units and vocabulary upon which sentence construction and communicative discourse are based. Listening comprehension may well be enhanced by the new technology, if audio and video cassettes tied to computer-assisted programs can offer listening comprehension practice, along with writing practice. Performance studies using the new media will help language educators decide whether the investment in the new technology is worth the time and effort.

Table I

Data: Computer and Non-Computer Groups

Non-Computer Group ($\bar{n}$=13)

ID	Age	Sex	Treatment	Final	SAT	Experience	MLA
103	23	0	0	50	104	0	12.7
105	20	0	0	76	108	4	16
108	20	0	0	57	107	0	10
110	19	1	0	89	108	6	20
112	22	1	0	78	78	0	8.7
113	21	0	0	88	115	4	24
114	22	0	0	58	103	3	16
117	20	1	0	81	105	5	21
118	19	1	0	70	68	0	6
202	19	1	0	63	108	4	11
204	20	0	0	61	73	0	5
210	19	0	0	71	105	4	11.7
214	18	1	0	85	94	4	20
$\bar{M}$=20.15		.46	0	73.21	98.15	2.61	14.03

Computer Group ($\bar{n}$ = 12)

ID	Age	Sex	Treatment	Final	SAT	Experience	MLA
101	19	1	1	78	95	0	6.7
102	19	1	1	77	100	0	5.3
104	18	1	1	76	104	4	16
109	23	0	1	80	103	0	11
116	18	1	1	83	63	4	15
203	18	1	1	99	112	0	23
206	21	0	1	79	144	0	12.7
207	21	1	1	92	109	4	33.7
209	22	0	1	79	104	4	13.3
211	20	1	1	70	105	0	15.7
212	23	0	1	79	98	7	13
215	21	1	1	89	104	0	30.3
$\bar{M}$=20.25		.67	1	81.75	103.42	1.92	16.31

Table II

Group Differences for Sex, Prior French, SAT, MLA, and Final Exam Scores

	Non-Comp. Group $\bar{M}$	Comp. Gp. $\bar{M}$	F - Value	2 Tailed Prob.
Sex (%F)	46.15	66.66	1.11	.869
Experience	2.59	1.85	1.23	.235
SAT scores	101.41	98.15	103.42	.588
MLA scores	14.03	16.31	2.04	.235
Final exam	71.31	81.75	2.59	.125

*statistically significant at the .05 level, 2-tailed test

Table III

Analysis of Covariance of Final Exam Scores with Scholastic Aptitude Test Scores (SAT) and Prior Experience as Co-Variates

	Sum of Squares	df	Mean Square	F	Sign. of F
Covar.	357.156	2	178.578	1.641	0.218
SAT	7.201	1	7.201	0.066	0.800
Exp.	336.185	1	336.185	3.088	0.093
Treat.	766.381	1	766.381	7.041	0.015*
Expl.	1037.577	3	345.859	3.177	0.045
Resid.	2285.863	21	108.851		
Total	3323.440	24	138.477		

*statistically significant at the .05 level, 2-tailed test

Table IV

Analysis of Covariance of MLA Listening Comprehension Scores with SAT Scores and Prior French Experience as Co-Variates

	Sum of Squares	df	Mean Square	F	Sign. of F
Covar.	305.837	2	152.919	3.374	0.054
SAT	118.959	1	118.959	2.625	0.120
Exp.	155.882	1	155.882	3.439	0.078
Treat.	31.728	1	31.728	0.700	0.412
Expl.	328.206	3	112.735	2.487	0.080
Resid.	951.800	21	45.324		
Total	1290.006	24	53.750		

*statistically significant at the .05 level, 2-tailed test

References

Corbett, S. & Smith, W. (1984). Identifying student learning styles: proceed with caution. Modern Language Journal, 68(3), 212-221.

Hope, G. (1982). Elementary French computer-assisted instruction. Foreign Language Annals, 15, 343-353.

Lalande, J. (1984). Reducing composition errors: an experiment. Foreign Language Annals, 17(2), 109-118.

Reinert, H. (1976). One picture is worth a thousand words: not necessarily! Modern Language Journal, 60, 160-68.

Sanders, D., & Kenner, R. (1983). Whither CAI? the need for communicative courseware. System, 11(1), 33-39

Suppes, P. (1965). Computer-assisted instruction in the schools: potentialities, problems, prospects. Psychology Series Technical Report. Paper presented at the Scientific Computing Symposium, from Eric Abstracts, 1981, Report No. Su-IMSS-TR-81.

Taylor, H. (1981). Computer assistance in the German individualized instruction program at the Ohio State University. In D. Harris and L. Nelson-Heern (Eds.) Proceedings of NECC 1981 Conference. Iowa City, IA: University of Iowa.

A Comparison of Foreign Language Learning with Computer-Assisted and Non Computer-Assisted Study

The purpose of this study was to develop seven diskettes of computer software to accompany the beginning French course at the University of Idaho and to examine the efficacy of students' reviewing French for written quizzes and the final examination using this method of review rather than the traditional method of review. The software was designed to conform to the types of exercises found in the textbook used in class and closely approximated the workbook exercises which all students were required to complete. The design resembled the "drill and practice" lesson structure but also used string-searching algorithms and other pedagogical tools to make the process more individualized and communicative.

The American Council on the Teaching of Foreign Languages has provided a widely-accepted model for the definitions of language proficiency skills, and these definitions were related to the course goals of French 101. A common final examination reflecting these goals for writing proficiency was designed and used as the dependent variable.

The subjects for this study were university students registered in two sections of French 101 using the same instructor. Initial background data were gathered via questionnaire at the beginning of the semester. After a brief introduction to the APPLE computer laboratory and the software, students were asked if they would like to "test" the software. It was emphasized that they would not receive a grade or any extra benefits for this activity, but that it could offer supplementary practice for written examinations. Each student was asked to fill out an evaluation of each diskette after reviewing the software. These were collected by lab assistants and given to the instructor after the completion of the course.

After the completion of the final examination and the course, background data on the students' general academic achievement potential as measured by the Scholastic Aptitude Test or the American College Testing Service's test, the students' age and sex, and their prior experience in French were entered. The data were analyzed using the analysis of covariance. Holding the variables of prior exposure and general ability constant, the two methods of review were contrasted and their effect determined on the dependent variables, the final examination in written French, and the Modern

Language Listening Comprehension test.

RELATIONSHIPS AMONG KNOWLEDGE STRUCTURES AND INFORMATION IN A COMPUTER-BASED PROBLEM SOLVING ENVIRONMENT

Elise G. Millikin

University of Cincinnati

Knowledge structure development has been a fertile area for research in domain-free as well as domain-specific problem solving. Before approximately 1975, research was based on a variety of stepwise instructional models that generally directed students through a linear process in inquiry. Scientific research or intuitive thinking during the problem solving process rarely proceeds in the linear fashion indicated in the models. Instead, the novice problem solver uses trial and error guided by a knowledge development strategy.

Results of research in an information processing paradigm have shown that (1) interactions (relationships) exist between learners' knowledge structures and requests for information (Mayer, 1975); (2) availability of relevant information influences learning outcomes (Morris, Bransford & Franks, 1977); (3) knowledge and skills that are high in a hierarchy of learning skills are better remembered and form an organizing framework (Fredericksen, 1975; Gentner, 1976; Meyer, 1980); and, (4) processing of information required for different kinds of learning interacts with type of material resulting in qualitatively different knowledge structures in learners (Mayer & Greeno, 1972).

As a result of advances in cognitive psychology, learning systems have been developed that are based on generative hypotheses which conceive of learners as being actively engaged in constructing or generating meaning from material. Because the computer offers instructional designers capabilities of interactivity and options to vary the amount or extend learner control, theories of generative processing of materials by the learner can be experimentally studied. Furthermore, archival methods have made available statistics of student use, and provided information on data and information requests.

Models of intelligent teaching systems based on mastery learning concepts have been designed to control learner responses to a great extent and have restricted the amount of information obtained regarding learner/information interactions. A computer-based problem solving environment that allows for learner controlled access to different forms of information, and that unobtrusively records learner access of information as well as pre- and post-test scores, may allow researchers to discover new relationships between learning and the need for different types of information as they attempt to solve problems.

Theoretical Background

Four distinct models of learning can be identified as having stimulated problem solving research. Research based on the Gestalt model was mainly concerned with perception and investigated cognitive processing in terms of whole and part-whole relationships. In the behavioral model. application of the interpretative framework of learning theory to problem solving generated empirical research. Research based on a psychometric model focused on factor analysis of individual differences and incorporated the structure of cognitive processes; and, the information processing framework has incorporated systems theory and artificial intelligence models.

These four approaches to studying problem solving have resulted in contributions to the study of problem solving using computers. The major contribution of the Gestalt approach has been a focus on perception. Perception from a Gestalt position can be described in terms of the appearances of things that are relationally determined. That is, what is seen, felt or heard is determined not by individual properties of things, but by the temporal, spatial, and other relations between them.

Even though explanations (by behaviorists) of relationships between psychological mechanisms taking place during the problem solving process and the task are not possible within the S-R framework, behaviorist concepts such as trial and error behavior, hierarchies of behavior and generalization have considerable relevance as explanations of problem solving. Contributions from research based on psychometric models have included factor analytical techniques that offer descriptive results to aid in planning of investigations and facilitation of the selection and assessment of sets of variables for further investigations. Information processing models have provided a language which represents human performance in terms of cybernetic concepts and an effective tool for the analysis of behavior sequences and systems.

The work of Newell and Simon has provided a basis for the construction of the information processing theory. The theoretical framework is based on the importance of two dimensions of the problem solving process-demands of the task environment and psychology of the subject. The theory postulates that the subject's response to a problem is determined by his or her perception of the stimulus and the feedback received from the environment as a result of behavior. George (1980) incorporates this postulate in his model of problem solving behavior.

George's (1980) model of problem solving behavior is based on a theory of cognition which is premised on the following: (1) knowledge is composed of the total set of beliefs that we hold - some false, some true; (2) beliefs are the source of actions; and, (3) beliefs, when activated, give rise to expectancies. Which beliefs are converted into expectancies depends upon previous associations between certain stimuli and certain beliefs. This model takes into account internal and external states of motivation which are derived from drive and urgency of the subject. Motivation operates to determine speed of selection and speed and strength of response elicitation. Cues, signs, symbols, language, meaning and truth are used to categorize and identify stimuli and possible outcomes.

Related Research

Computer based research has proceeded in two directions: (1) simulation of the human subject as a problem solver; and (2) discovery of efficient algorithms for the solution of tasks. Included in the last category is research in specific subject matter areas, which was of interest in this study. In science education research that has addressed problem solving behavior, there have been attempts to identify correlates to successful problem solving behavior. Chi (1981) concluded that domain-specific knowledge deficiencies limited the range of inferences generated by novices.

If knowledge deficiencies limit problem solving ability, it is appropriate to focus on how learners process different forms of information. A concern of the information processing theorists relates to the form with which information is received, stored and transmitted. What appears to be ignored is that the nature of the information processing and tne structure of the information might be expected to have decisive effects on the facilitation of learning outcomes. The artificial intelligence community has categorized knowledge as conceptual, procedural and imaginal. Imaginal knowledge seems to work more like perception than a network of facts; but has usually been ignored in AI research.

Results from visual imagery research include the ideas that images and propositions exist at different levels in memory and that visual information is represented on a relatively superficial level, but translated into and stored as propositions at deeper levels (Finke & Kosslyn, 1980; Randhawa, 1981; Rosenberg & Simon, 1977). Kosslyn (1980) offered evidence against the idea that images are ultimately represented as propositions and suggested that further evidence is needed in which discrimination is not based on propositional lists.

The processing of visual information entails its assimilation to existing schemata, the modification of schemata to accommodate it, and the translation of analogs or images in short-term memory into more abstract forms at high levels that allow assimilation and accommodation to take place (Finke & Kosslyn, 1980. The interaction of schemata with perceived information poses the question of levels of information processing, initial stages involving psychophysical and perceptual processes that are shallower in terms of encoding levels than the deeper schematic levels (Randhawa, 1971). These findings support George's ideas of associations, categorizations and sets of beliefs.

Applying imagery hypotheses to problem solving, Paivio (1979) stated that individual differences in mental imagery imply that people differ in their approach to problem solving. Kaufmann (1983) found that visual imagery is a significant factor in creative problem solving and that visual presentation of tasks facilitated problem-solving performance, which was interpreted as a confirmation of the visual imagery hypothesis.

Conceptual knowledge is represented in AI systems using semantic networks which consist of nodes representing concepts and links representing the relationships between concepts. Semantic networks in instructional applications represent the subject matter to be learned. Newell and Simon (1972), and Rumelhart, Lindsay and Norman (1972), have studied how knowledge of academic subject matter material or problems are represented in memory. These studies have produced models that describe the nature and characteristics of knowledge structures. Other researchers (Champagne et al, 1981; Jonasen, 1984) have attempted to develop measures of knowledge structures, trace changes in structure as a result of instruction, and explore the relationships between knowledge structures and problem solving.

For example, Champagne et al (1981) incorporated the structure of geology concepts derived from experts' writings, and from subject matter material. He compared students' structural representations of concepts before and after instruction, and concluded that students' own graphical/semantic information representations were valid measures of knowledge structures. Jonasen (1984) who has worked with pattern notes (best described as spatially arranged word associations) suggested that they offer an alternative to assessing prior knowledge and cogntive structure. However, training is required of the subject to efficiently organize and represent concepts, relationships and procedures. Questions based on a taxonomy and mapped might offer an alternative to assessing learners' knowledge structures through measures that require training and provide a familiar format. Russell and Chiapetta (1981) used questions based on Bloom's taxonomy of educational objectives to measure students' knowledge and thinking skills after an instructional treatment that included a problem solving strategy. They concluded that instructional strategy does have effects on knowledge structures and on the application of knowledge to higher level cognition and problem solving.

Research needs to address the questions of cognitive processes and instructional strategies simultaneously (Winn, 1982), because learning is influenced as much by instructional strategy as it is by mental skill; given that other variables such as knowledge of task, form in which information is presented, and general ability all influence learning. Winn (1982) summs up the relationships between cognitive processes and instructional strategies.

1. Learning occurs when learning strategies are meaningfully brought to bear on learning tasks. Learning strategies, mental skills and cognitive processes can be controlled by means of instructional strategies which are either embedded in or detached from the material to be learned.
2. Information is represented internally as schemata, which may or may not contain realistic representation of reality. Learning involves the assimilation of new information into these schemata, and the accommodation of schemata to it.
3. Visual processes and strategies are exemplified by imagery and feature recognition where properties of stimuli, such as form, spatial arrangement, and so on, influence how information is processed.
4. Visual processes include perceptual processes, assimilative processes which involve recognizing and integrating information into schemata, and analogical processes which involve temporarily assimilating new information into abstract schemata.

Model and Technique Development

A system was developed based on cognitive theories of learning and problem solving. The system model was based on Lazzaro & Szabo's (1976) model which is independent of the subject material and the way in which the material is learned. A general description of the aims of the package follows. A problem in plate tectonics will be presented that will require that the learner understand related concepts, make hypotheses regarding the solution to the problem, and decide on an appropriate solution. In the process of solving the problem the learner may request from the system visual and textual information. Appropriate clues to the information held in the system were provided. The student is in control of the secreen sequence at all times except during the pre- and post-tests. The remainder of this paper will describe the construction and assimilation of components and specifics of instructional design used.

The production of the system generally followed the guidelines for developing educational computer programs set out by Fox and Rushby (1978). The stages included: (1) functional specifications of the model and requirements of the system;, (2) system and program specification and coding; and, (3) validation of the system components. Much like the National Development Programme in Computer Assisted Learning project, one of the aims was to develop an unobtrusive system in which the data is captured by the system during interaction. Transferability and adaptability of the model were also project objectives.

The constants were defined as the problem statements, the textual fields, the visual (graphic) fields and the pre- and post test questions. Complexity of problem was determined by the number of components and by the problem solver's familiarity with the context of the problem (Steinberg, 1983). Solving a problem can mean getting the right answer (product) or generating a strategy (process). In this research, generating a context-specific strategy was considered.

Chovan (1985) suggested that the process of knowledge extraction from any source is done in three steps; (1) problem definition, (2) implementation and refinement of knowledge and scope or domain, and (3) representation of facts and relations in the final product. The problem was defined in collaboration with an eighth grade science teacher who stated that his students had trouble comprehending plate tectonics. Software databases and catalogs were searched for possible programs in plate tectonics. No programs in plate tectonics were found at the eighth grade level. The teacher then assisted in defining the goals and limiting the scope of the problem which included delineating concepts that eighth graders should be able to understand.

Second, sources of expertise, besides the science teacher were located. Eighth grade science textbooks, college level textbooks, library references, geophysics graduate students and professors, and a week-long seminar in plate tectonics were used as sources of expertise. Printed material was perused for background information and jargon; content was evaluated and composed. At this point, the eighth grade teacher continued to help restrict the domain of the problem and the geophysics experts helped to refine the content information.

The final stage was representing facts and relationships, validating it further for content, and then revising it. In the final stage, some relationships were added and some insignificant ones were deleted. To validate content in the learning program, a formative evaluation procedure for assessing instructional materials was followed. Subject matter experts who had expertise in the area of plate tectonics, were university level professors, and who had not previously participated in the refinement of the content representation, were identified and interviewed in a systematic fashion. They were given a copy of the content background, structure and definitions, the textual and visual fields and the test questions. They were asked to respond to the following questions: "Is the concept stated clearly?" "Are there any parts that should be eliminated or added to the concept?" "Do the graphics clearly portray what is stated?" After two weeks, an interview was scheduled with them to discuss their comments. Discussions of instructional technique were avoided. After all responses were collected and summarized, appropriate revisions of the content components were made.

As suggested by Hall (1982), Bloom's hierarchy of learning objectives was used as a basis to write test items to evaluate knowledge structures. Levels of items were knowledge, comprehension, application, analysis and synthesis. Test questions were constructed according to guidelines suggested by Noll and Scannel (1972). Initially eighty-one questions were written, thirty at the knowledge level, ten at the comprehension level, eleven at the application level, ten at the analysis level, and ten at the synthesis level. To validate the content, three subject matter experts (SME's) reviewed the questions for clarity of content and for determination of clear answers and non-answers. After the questions were revised according to the SME's suggestions, three experts in learning theory and question construction were asked to respond to the following questions for construct validation:

1. Does the stem clearly and distinctly identify the task posed by the item?
2. Are the answer alternatives stated with the same clarity and preciseness as the stem?
3. Are the choices in parallel form?
4. Is the length of the item and the choices appropriate for the purpose of the item?
5. Does the question fall within the appropriate (named) level?

Responses from the test construction experts were summarized and revisions were made. The final version of the test consisted of thirty items at the knowledge level, eight items at the comprehension level, eight items at the application level, eight items at the analysis level, and eight items at the synthesis level.

A data trail to record learner access to the textual and visual information databases, time spent with the problem, and pre- and post-test responses was designed. Variables and permitted ranges for each variable were defined.

Program procedures were written in Structured English and later translated to pseudocode. A modular design of programming was employed which allowed the programmer to divide the problem into smaller, simpler pieces which were tested independently. The programmer used the C language and structured programming methodology which lent structure and discipline to the program form, design, coding and testing.

Design principles were based on cognitive theories of learning (Winn, 1984). Specific attributes of interactive learning with computers (Jay, 1984) were considered in the design and development process.

1. First Principle. Limitations occur for number of tasks and level of difficulty that individual learners can accommodate in memory and attention. Implementation of this principle occurred in the following ways. One idea per screen was presented using terminology and number of sentences required (determined by SME's and the designer) to sufficiently explain the idea. The learner determined the amount of time needed to read and respond to textual material because control of time was given to the learner, making timing self-paced and flexible. Prompts were utilized to eliminate scrolling, an unnatural format. The learner was kept informed in long delays (more than two seconds). Because inductive questions were used for the problems, learners were given approximately forty minutes to solve; and supplemental materials for answering and working toward a solution were provided since the task was complex. Learners were given a paper and pencil format for problem solution even though the problem was presented on the screen. This was done so that training for sophisticated interaction with a diagram or word processing format could be eliminated.
2. Second principle. Elaborate semantic processing (deeper processing levels) produces better long-term memories; and the greater the task demands, shorter time periods are needed to keep performance at a satisfactory level. The major divisions of the system (modules) lasted for 15-20 minutes each, i.e. pre-test, problem, post-test. The opportunity to exit during the problem stages was provided.
3. Third principle. Attention should be paid to language or text characteristics, i.e. content, style, syntax, vocabulary, the type of user (age) and the intention of the message (provide information, questions, or prompts). New words were defined and jargon, unnatural codes and humor was avoided. First person form was used in order to achieve friendly communication. Consistency in presentation and expected input was followed by using language that is graphic in content and style, by using concrete terms when possible, by using multiple presentation formats, and by using simple, concise syntax. The message required the user to think or process information that was related to other information in other screens and ideas. Materials were developed in the natural dialogue and inquiry method so that learners could peruse categories and subcategories of ideas that related

to information he/she perceived to be valuable.

4. Fourth principle. Graphics and visual processing helps promote understanding of textual material and enhances memory and motivation. This principle was followed in the following ways: Color was used to code, differentiate or highlight; and to direct attention. Cuing was used in the form of arrows, titles, lines, and hue to direct attention or distinguish features. Graphic descriptions were used in text format to describe abstract ideas. Pictures, maps, diagrams and drawings used in conjunction with text were used to lessen memory demands and enhance learning. Graphics (special characters, underlining, enlarged text, and color) to enhance information. Use of mnemonics by visualizing the concepts increases memory; therefore, each visual depiction was closely representative of its textual counterpart. Color use agreed with habitual denotations, was used to teach relationships and differences in a display, and was used to denote similar functions or properties.

5. Fifth principle. Learners require feedback; rewards should be natural and variable; and, hints should be used to sharpen a closer response. The user was informed whenever performance was scored or evaluated (only pre- and post-tests). The user was informed when material was being searched for or when calculations were being made. Neither rewards nor hints were used in this experiment. More than one opportunity was given for answering questions and the students was allowed to go on if he/she could not derive the answer. Poorly worded or ambiguous questions were eliminated and incorrect responses were chosen carefully and validated.

6. Sixth principle. Models of learners include explicit and implicit assumptions based on the principle that processing abilities mature with age; and that children differ in abilities to think abstractly, hypothetically, deductively, or logically. Presentations were designed to be appropriate for eighth graders' level or state of knowledge; and the problem presented stimulated the innate quest to explore the world by using inquiry techniques. Hypothetical deductive, and problem-solving reasoning was incorporated and external strategies, such as cues and directions were utilized. Because learners differ in the effective learning of associative information, control of time and retrieval of information was given to the learner.

The final result of these development procedures is a computer-based system for human problem solving that stores background information accessible to the learner through menus. Perhaps empirical data gained from research on a CAI of this type will contribute to an understanding of the types of knowledge and processes that are involved in problem solving and the relationship between knowledge structures and need for information. The incorporation of pre- and post-tests will allow for the evaluation of knowledge structures and the data trail will allow for the collection of learner interactions with the information database and the problem. Also contributions to instructional design theories based on cognitive theories of learning can be made. Finally, the structure of this system has capabilities to incorporate intelligent components for directing learning or suggesting strategies when appropriate and for interactive simulation.

References

Champagne, A. B., Klopfer, L. E., Desena, A. T. & Squires, D.A. (1981). Journal of Research in Science Teaching, 18(2), 97-111.

Chi, M. T. (1981). Expertise in problem solving. Pittsburg University, PA: Learning Research and Development Center.

Chovan, J. D. (1986). Prototype tools for information extraction to be used in the creation of computer-based training materials. Paper presented at the annual convenction of the Association for the Development of Computer-based Instructional Systems, New Orleans, January.

Drake, C. L., Imbrie, J., Knauss, J. A. & Turekian, K. K. (1978). Oceanography. New York: Holt,Rinehart and Winston, 298-320.

Finke, R. A., & Kosslyn, S. M. (1980). Mental imagery acuity in the peripheral visual field. Journal of Experimental Psychology: Human Perception and Performance, 6, 126-139.

Fox, J. & Rushby, N. (1978). Guidelines for developing ducatinal computer programs. Computers & Education, 3, 35-41.

Fredericksen, C. H. (1975). Acquisition of semantic information from discourse: Effects of repeated exposures. Journal *of Verbal Learning and Verbal Behavior*, *14*, 158-169.

George, L. H. (1980). *Problem solving*. London: Duckworth.

Gentner, D. R. (1976). The structure and recall of narrative prose. *Journal of Verbal Learning and Verbal Behavior*, *15*, 411-418.

Hall, K. A., & others (1981). *Taxonomy of instructional strategies for computer based education*. Columbus: Ohio State Univ. & Control Data Corporation (ERIC Document Reproduction Service No. ED 217 867).

Jay, T. B. (Jan., 1983). The cognitive approach to computer courseware design and evaluation. *Educational Technology*, 22-26.

Jonasen, D. H. (1984). Developing a learning strategy using pattern notes: A new technology. *PLET*, *21*, 3, 163-175.

Kaufmann, G. (1979). *Visual imagery and its relation to problem solving: A theoretical and experimental inquiry*. New York: Columbia University Press.

Kosslyn, S. M. (1980). *Image and Mind*. Massachusetts: Harvard University Press.

Lazarro, A. & Szabo, M. (1976, April). *Problem-solving behaviors as a function of instructional mode, conceptual systems and field independence in a CAI environment*. Paper presented at the annual meeting of the National Association for Research in Science Teaching, San Francisco, CA (ERIC Document Reproduction Service No.1 ED 123 131).

Mayer, R. E. (1975). Information processing variables in learning to solve problems. *Review of Educational Research*. *45*(4), 525-541.

Mayer, R. E. & Greeno, J. G. (1972). Structural differences between learning outcomes produced by different instructional methods. *Journal of Educational Psychology*. *63*, 165-173.

Meyer, B. et al (1980). Use of top-level structure in text. *Reading Research Quarterly*, *16*, 72-103.

Morris, C. D., Bransford, J. D. & Franks, J. J. (1977). Levels of processing versus transfer appropriate processing. *Journal of Verbal Learning and Verbal Behavior*, *16*, 519-533

Noll, V. H. & Scannell, D. P. (1972). *Introduction to educational measurement, 3rd ed.* Boston: Houghton Mifflin Co., 191-236.

Rowe, H. A. H. (1985). *Problem solving and intelligence*. Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers.

Paivio, A. (1979). *Imagery and verbal processes*. Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers.

Randhawa, B. S. (1971). Intellectual development and the ability to process visual and verbal information. *AV Communication Review*, *19*(3), 298-312.

Rosenberg, S. & Simon (1977). Modelling semantic memory; Effects of presenting semantic information in different modalities. *Cognitive Psychology*, *9*, 293-325.

Rumelhart, D. E., Lindsay, P. H., & Norman, D. A. (1972). A process model for long-term memory. *Organization and Memory*, ed. E. Tulving & W. Donaldson. New York: Academic Press.

Russell, J. M. & Chiapetta, E. L. (1981). The effects of a problem solving strategy on the achievement of earth science students. *J. of Research in Science Teaching*, *18*(4), 295-301.

Steinberg, E.R. (1983). Problem complexity and the transfer of strategies in computer-presented problems. *American Educational Research Journal*, *20*(1), 13-28.

Winn, W. (1982). Visualization in learning and instruction: A cognitive approach. *ECTV*, *30*(1), 3-25.

ADCIS Proceedings, November 1986

The Effects of Textual and Animated Orienting Activities and Practice on Learning from Computer-Based Instruction

Lloyd P. Rieber, The Pennsylvania State University

Michael J. Hannafin, The Pennsylvania State University

Abstract

In this study, the effects of textual versus animated orienting activities and practice on the learning of elementary science material was studied. Elementary school students were randomly assigned to either a Text, Animated, Text plus Animated orienting activity group or a control group having no activity, and to either a Practice of No Practice group. Upon completion of the lesson, students were administered a posttest measuring both rule-using and problem solving. Results indicated that the nature of the orienting activity did not affect the learning of either rule-using or problem solving. However, a negative effect was found for practice.

The science of instructional design has become increasingly interesting and complex. Recent interest in instructional design has focused on the potential of computer technology to deliver effective and efficient instruction. Much research has shown that carefully designed instruction can be delivered effectively via computer (see, for example, Edwards, Norton, Taylor, Weiss&Dusseldorp, 1975; Bangert-Drowns, Kulik,&Kulik; 1985; and Kulik, Kulik,&Bangert-Drowns, 1985).

Certain features of computer technology are well-suited to the needs of instructional designers. For example, instruction can be delivered repeatedly across learners without random variation. Other features, such as the ability to branch to appropriate lesson segments based on individual learner needs and to provide appropriate and immediate feedback can be easily incorporated into computerized instruction. Lesson designers can alsc incorporate a wide range of teaching strategies and activities into computer-based instruction (CBI).

One goal of CBI designers is to utilize capabilities presumed to be unique to the computer. Though controversy exists as to the unique effectiveness of the computer as an instructional tool (see, Petkovich&Tennyson (1984) vs. Clark (1983); Clark, 1985), certain capabilities offer interesting instructional potential. Animation is one such capability which can be more easily integrated into computerized instruction than other instructional systems. For the purposes of this paper, animation is a series of rapidly changing computer screen displays that presents the illusion of movement (Caraballo, 1985). Computerized instruction has typically used animation as an extrinsic motivator involving such things as cartoon figures acting as feedback to student's responses. However, animation can also be used to support or enhance instruction directly and indirectly. An example of instructionally supportive animation is the animated block accelerating down an inclined plane to teach a concept in a physics class. (See Alesandrini, 1984, for a discussion of the use of visuals in instruction.)

Many perspectives on the design of effective instruction exist. Recent interest has shifted to the influence of cognitive processes during instruction (Clark, 1984; Gagne&Dick, 1985; Hannafin& Rieber, 1986). CBI designers attempt to strengthen learning by encouraging individually relevant cognitive processes through various lesson activities or events (Gagne, Wager, &Rojas, 1981).

Recent attempts to improve learning through the use of orienting activities have also been reported. An orienting activity is a mediator through which new information is presented to the learner (Hannafin&Hughes, 1986). Orienting activities comprise advance organizers, pre-instructional objectives, pre-questions, and other similar techniques (see, for example, Allen, 1970; Alvermann, 1981; Ausubel, 1960; Kaplan&Simmons, 1974; Mayer, 1979,1984; Walsh&Jenkins, 1973). Interstingly, the effectiveness of orienting activities during CBI has been inconsistent (Hannafin, Phillips, Rieber, &Garhart, 1986).

Recent research has demonstrated the power of practice to override effects normally expected from orienting activities (see, for example, Hannafin, 1986; Hannafin, Phillips,&Tripp, 1986). Researchers have speculated that practice often subsumes effects expected from orienting activities alone. This is consistent with the guidelines provided by Mayer (1979), who suggested that advance organizers are often unnecessary in the presence of well-designed instruction. In certain cases, practice activities may also orient learners to subsequent instruction (Phillips, Hannafin,&Tripp, 1986).

The purpose of this study was to examine the effects of different orienting acticities and practice on the rule-using and problem solving skills of elementary school students. It was hypothesized that students provided orienting activities containing textual and animated information would acquire both skills more effectively than either activity alone or neither activity. It was further hypothesized that practice would be of greatest value where the orienting support was minimal.

Methods

Subjects

The subjects consisted of 111 fourth, fifth, and sixth graders from a rural public elementary school. These subjects represented approximately 80% of the student population of these grades. Participation was voluntary and selection based on parent consent. The subjects represented a typical cross-section of students typically found in an elementary school. The subjects consisted of 56 girls and 55 boys. The proportion of subjects across grade levels were 24 fourth grade, 41 fifth grade, and 46 sixth grade.

CBI Lesson Content

The CBI lesson described and explained Isaac Newton's Law of Motion. The lesson material was divided into four parts: 1) introductory material; 2) motion resulting from equal forces in opposite directions in one dimensional space; 3) motion resulting from unequal forces in opposite directions in one dimensial space; and 4) motion resulting form equal and unequal forces in two dimensional space.

The first lesson part introduced the learner to Isaac Newton's formal discovery of certain physical laws. This part also initiated the sequence of learning activities used for the rest of the lesson. Most of the information presented in the first part was factual in nature. The second part introduced the concept that equal but opposite forces are needed to cause objects to stop. This section dealth only with one-dimensional space. The third part expanded this notion to include the effects of unequal forces acting upon a stationary object. The final direction and speed of an object is a combination of all of the forces from both directions. The third part also presented these concepts in one-dimensional space. Finally, the fourth part added the notion of two-dimensional space to the above concepts. Again, the final direction and speed of the object results systematically from the sequence of forces which acted upon it.

Each of the four lesson parts included the same two instructional segments in the same sequence: appropriate orienting activity and instructional frames. All instruction was presented at an introductory level with the technical descriptions of the forces of gravity and friction removed. Approximately 30 minutes was requred to complete the lesson.

Lesson Versions

Each lesson included one of four orienting activities. Each orienting activity was controlled and paced externally and was presented immediately prior to each lesson part. Each orienting activity was presented for approximately one minute.

Text. In this orienting activity, a text-only computer screen prompted the learner to read a one sentence summary of th basic physical science concept to be discussed in the upcoming lesson part. An example of a text orienting activity is: "Read the following information carefully. It will help you in the next section. In order for a moving ball to completely stop, all forces must be EQUAL in all direction."

Animation. In this orienting activity, the learner was prompted to watch a short animation sequence. The animation sequence illustrated graphically a science concept without the use of textual information. An example of this activity is: "Watch the little ball below carefully. It will help you in the next section." A small ball was then kicked once to the right (a small arrow on the screen represented a kick) while ball movement was animated in a left to right fashion. When kicked once to the left, the ball stopped.

Text plus Animation. This orienting activity was the combination of the above two strategies. The text and the animation orienting activity described above were shown together on the same frame while the learner was prompted to read and watch the information carefully.

No Activity. In this activity, the learner was given no information about the upcoming lesson. However, the lesson paused for roughly the same time required for the orienting activities while the following prompt was given: "The computer will be busy for about a minute. Please think about what you've read so far while you wait."

Practice. Learners were give a variety of activities to rehearse the lesson concepts. for example, one practice activity displayed a series of left and right arrows representing a sequence of kicks. The learner was then asked to choose which of four given outcomes would best describe the final motion of the ball. After responding, appropriate feedback in the form of knowledge of correct results was provided while the computer animation of the kick sequence was presented. During another practice activity, the ball was animated according to a predesigned pattern. The learner was then asked to choose which of four given kick sequences best described the motion of the ball. After responding, the learner was given appropriate knowledge of correct results. Similar practice activities were used throughout the lesson.

No Practice. In this version, students received no practice after the presentation of the lesson information. After the appropriate lesson presentation, the learner was routed directly to the orienting activity of the next lesson part.

Lesson Posttest

The 24 item posttest consisted of two types of questions: rule using and problem solving (Gagné, 1977). A total of 12 rule using questions and 12 problem solving questions were included in the posttest.

Design and Data Analysis

A 4 X 2 factorial design was used. Four levels of orienting activities (Text, Animation, Text plus Animation, None) were crossed with two levels of practice (Practice, No Practice). In addition to the independent variables of orienting activities and practice, an overall achievement grade equivalent score of a standardized achievement test, was used as a covariate. This score was used in order to minimize the effects of an unbalance of students from different grade levels in the various treatment combination.

Procedures

Subjects were randomly assigned to one of the lesson versions and the lesson was presented accordingly. Assignment was proportionally stratified to ensure balance of subjects from achievement levels in each lesson version. All instruction and testing was administered by computer in a lab containing microcomputers. Upon completion of the lesson, students were prompted to inform the proctor. The posttest was given to the students immediately upon completion of the lesson.

Results

Table 1 contains adjusted means for each treatment combination. No significant differences were found for orienting activities, $F(6,204)=1.13$, $p<.05$. Mean scores were comparable irrespective of the orienting activity provided. A marginally significant effect was found for practice $F(2,101)=3.668$, $p<.05$. However, the direction of this effect was not predicted. Practice actually hampered performance for both rule using and problem solving skills. No interactions were detected.

Table 1.

Adjusted Means for Rule-Using and Problem Solving Questions

Embedded Activity		Orienting Activity Text	Animated	Text + Animated	None
		Rule-Using			
Practice	Mean	4.84	5.45	4.40	4.86
	n	14	14	14	14
No Practice	Mean	6.13	4.86	5.08	4.24
	n	14	14	14	14
		Problem Solving			
Practice	Mean	5.39	4.86	4.29	5.14
	n	14	14	14	13
No Practice	Mean	5.92	5.22	6.22	6.06
	n	14	14	14	14

Discussion

The purpose of this study was to examine the effects of text-based and computer animated orienting strategies and practice on rule-using and problem solving. The results suggest that orienting activities, whether text-based or animated, do not exert particularly powerful influences on learning.

The lack of differential effects attributable to orienting activities during computer-based instruction was consistent with several recent studies involving computer-based learning (Hannafin, Phillips, Rieber&Garhart, 1986; Hannafin, Phillips,&Tripp, in press; Hannafin, 1986). However, the activities used in the present study were believed to be consistent with the capabilities of the computer and were believed to provide powerful orienting mechanisms. The textual orientation provided a verbal representation of the science concept while the animation provided a mental image of each particular science concept.

Cognitive psychology researchers have posited mental imagery and verbal representations as primary mental stuctures. Many researchers have noted the power of pictures over abstract words in stimulating higher levels of retention (see for example, Bower, 1972; Paivio, 1979). This research suggests that learners, especially young learners, are often better able to receive information in visual versus textual ways. When information is presented in highly verbal ways, young learners often create a mental image corresponding to the presented text. The implication is when instruction contains meaningful visual images to support lesson information, such as in the text plus animation treatment, learners should be in a better position to both store and retrieve the information to and from long term memory.

There are several explanations which may help to clarify why an orienting activity was not found. First, orienting activities, such as advance organizers, are probabley most useful when the subject matter lacks organization (Mayer, 1979). The instruction used in this study was carefully prepared and presented and may have subsumed potential learning gains derived from orienting activities. Also, it is possible that, due to the apparent difficulty of the lesson content, the students never fully elaborated the material at the time at encoding and therefore were less able to retrieve lesson information. Lastly, another possible explanantion comes from Carlson, Kincaid, Lance&Hodgson (1976) who noted that students tend to revert to their own individual strategies regardless of strategy prompting during instruction. Hence many students might simply disregard or ignore the potentially useful information or orienting strategies in favor of their own individual strategies.

The absence of the predicted interaction between orienting activity and practice was surprising. It was hypothesized that practice would decrease in in power as the potential meaningfulness of material was increased. This should also be true when children are provided with mechanisms to better encode information. It was predicted that the text plus animation activity would lessen the effect of the practice component since more elaboration cues were available at the time of encoding. The absence of the interaction is probably due to the ineffectiveness of the orienting activities already discussed: orienting activities simply did not make a difference in a fully supported CBI lesson.

Practice actually exerted a negative influence on learning in this study. This result is inconsistent with other findings and should be interpreted cautiously. It is evident from an analysis of the low posttest means that the lesson material was very difficult by the students. Providing additional instruction where heavy cognitive demands were already imposed upon the learner may be more detrimental than helpful to learning. The additional demands of practice may have created a type of cognitive overload. This phenomena is consistent with research from other areas of instructional design such as the use of visualization. Dwyer (1978), for example, found that although the use of pictures generally facilitated learning, this was not generally true when the material to be learned was too complex. Lesson designers should be cautious in the use of additional activities when lesson material is very complex and demanding.

Several directions for further research are indicated. Although the use of animation to enhance learning appears consistent with current theories of learning, recent studies have failed support this contention (see also, for example, King, 1975; Moore, Nawrocki& Simutis, 1979; A.Caraballo, 1985; and J. Caraballo, 1985). Continued research is needed to better define where the power of animation is useful and appropriate. Although the computer can deliver instruction in a variety of modes based on a variety of conditions, such conditions must be better defined. Additional research is needed to define optimal contexts for orienting activities and practice.

This study has raised several questions concerning how students process information, as well as how computer technology can contribute to this learning process. Based on this study, it appears that orienting activities are relatively insignificant in affecting learning when part of a well designed lesson. The negative effect of practice albeit small, demonstrates a need to use any instructional design component with caution and prudence. Future researchers should clarify and define superior instructional design models given the evolving capabilities of technology.

References

Alesandrini, K.L. (1984). Pictures and adult learning. Instructional Science, 13, 63-77.

Allen, D.I. (1970). Some effects of advance organizers and level of question on the learning and retention of written social studies materials. Journal of Educational Psychology, 61, 333-339.

Alvermann, D.E. (1981). The compensatory effect of graphic organizers on descriptive text. Journal of Educational Research, 75, 44-48.

Ausubel, D.P. (1960). The use of advance organizers in learning and retention of meaningful material. Journal of Educational Psychology, 51, 267-272.

Bangert-Drowns, R.L., Kulik, J.A., and Kulik, C.-L.C. (1985). Effectiveness of computer-based education in secondary Schools, Journal of Computer-Based Instruction, 12(3), 59-68.

Bower, G.H. (1972). A selective review of organizational factore in memory. In E. Tulving& W. Donaldson (eds.),Organization and memory. New York: Academic Press.

Caraballo, A. (1985). An experimental study to investigate the effects of computer animation on the understanding and retention of selected levels of learning outcomes. Unpublished doctoral dissertation, The Pennsylvania State University.

Caraballo, J.N. (1985). The effect of various visual display modes in computer-based instruction and language background upon achievement of selected educational objectives. Unpublished doctoral dissertation, The Pennsylvania State University.

Carlson, R.F., Kincaid, J.P., Lance, S.,& Hodgson, T. (1976). Spontaneous use mnemonics and grade point average. Journal of psychology, 91, 117-122.

Clark, R.E. (1985). Confounding in educational computing research. Journal of Educational Computing Research. 1(2), 137-148.

Dwyer, F.M. (1978). Strategies for improving visual learning. State College, PA.: Learning Services.

Gagne, R.M. (1977). The conditions of learning (third edition). Holt, Rinehart and Winston: New York.

Gagne, R.M.&Dick (1983). Instructional psychology. Annual Review of Psychology, 34, 261-295.

Gagne, R.M., Wager, W. W.,&Rojas, A. (1981). Planning and authoring computer-assisted instruction lessons. Educational Technology, 21(9), 17-21.

Hannafin, M.J. (1976). The effects of adjunct activities versus en-route practice on the learning of cued and uncued information. Submitted for publication.

Hannafin, M.J., Phillips, T.L., Rieber, L.R &Garhart, C. (1986). The effects of orienting activities and cognitive processing time on factual and inferential learning. Educational Communicatons and Technology Journal, in press.

Hannafin, M.J., Phillips, T.L.&Tripp, S.M. (1986). The effects of orienting and processing activities and practice on learning from interactive video. Journal of Computer-Based Instruction in press.

Hannafin, M.J.,&Rieber, L.P. (1986). The role of cognitive processing during computer-based instruction. Submitted for publication.

Kaplan, R.&Simmons, F.G. (1974). Effects of instructional objectives used as orienting stimuli or as summary/review upon prose learning. Journal of Educational Psychology, 66, 614-622.

King, W.A. (1975). A comparison of three combinations of text and graphics for concept learning. (Report No. NPRDC-TR-76-16). San Diego, CA: Navy Personnel Research and Development Center. (ERIC Document Reproduction Service No. ED 112 936).

Kulik, J.A., Kulik, C.-L.C., Bangert-Drawns, R.L. (1985). Effectiveness of computer-based education in elementary schools, Computers in Human Behavior, (1), 59-74.

Mayer, R.E. (1979). Can advance organizers influence meaningful learning? Review of Educational Research, 49, 371-383.

Moore, M.V., Nawrocki,L.,&Simutis, Z. (1979). The instructional effectiveness of three levels of graphics displays for computer-assisted instruction. (Report No. ARI-TP-359). Arlington, VA: Army Research Institute for the Behavioral and Social Sciences. (ERIC Document Service No. ED 178 057).

Paivio, A. (1979). Imagery and verbal processes. Hillsdale, NJ: Lawrence Erlbaum Associates.

Petkovich, M.D.&Tennyson, R.D. (1984). Clark's "Learning from Media": a critique. Educational Communications and Technology Journal. 32(4), 233-241.

Walsh, D.A.&Jenkins, J.J. (1973). Effects of orienting tasks on free recall in incidental learning. Journal of Verbal Behavior, 12, 481-488.

Contact: Lloyd Rieber
The Pennsylvania State University
Regional Computer Resource Center
302 Rackley Building
University Park, PA 16802

814-863-2427

ADCIS Proceedings, November 1986

PARTICIPATIVE MULTI-ROLE INSTRUCTION IN A DISTRIBUTED KNOWLEDGE-BASE ENVIRONMENT

Taracad R. Sivasankaran - Naval Postgraduate School

Ralph Ruby, Jr. - Arkansas State University

ABSTRACT

Computer-Based Instructional Systems of today do not have the interfaces to interact with other existing systems residing independently of them. When unexpected complex tutoring situations arise during an instructional session, it would be beneficial for current systems to cooperate with similar systems external to them, either in the same geographic location or otherwise, in order to draw upon additional sources of knowledge in a participative instructional mode and enhance the overall effectiveness of instruction. This paper presents a model representing the various roles and their interactions in a distributed instructional environment and proposes a knowledge-based design strategy towards building participative instructional systems. A prototype system in battleship management that employs the proposed model is also discussed.

1.0 Introduction

The use of computers as a tool in instruction and training is no new phenomenon. Present day computers utilize a wide range of techniques from simple electronic page turning to sophisticated interactive video disc simulations. However, many of the current computer-based instructional systems have two major set backs. They are, i) Standalone and ii) Algorithmic. Standalone is the machine-dependent feature of an instructional system which restricts its portability. Since computer hardware is not normally moved around geographically, the instructional system that runs on them also becomes location-dependent. The machine and locational dependencies have serious implications for future research in Computer-Aided Diagnostic and Prescriptive Instuction (CADPI). For instance, it is possible to imagine two instructional systems in overlapping or logically related domains of knowledge neither of which can solve certain complex problems independently, yet they can if only they cooperate and adopt a participative style in the problem solving. Such problems are commonplace in medical instruction where the instructor or the student might want to consult a specialist before a final determination is made. Thus, for synergy and effectiveness in overall instruction, it is essential that instructional systems in related fields be able to interact with one another in a symbiotic manner even though one may be external to the other and/or geographically dispersed (Sivasankaran and Ruby, 1986). The second major deficiency of current instructional systems is their characteristics of algorithmic execution (Simon, 1973). This implies that the system can run through only a set of paths predetermined by the system developer procedurally. Systems of this kind can have little, if any, heuristic reasoning, inferencing and learning abilities. But recent advances in network technology and artificial intelligence are beginning to offer methods by which the above two deficiencies can be redressed, if not totally eliminated.

The objective of this research is to build a model representing the various roles and their interactions in an instructional setting and identify a design strategy in building an integrated instructional system using distributed knowledge bases and teleprocessing networks. Section 2 of the paper covers the Network and Expert Systems technologies as applicable to the problems of integrating multiple instructional systems and their non-procedural heuristic based execution. Section 3 theorizes the different roles involved in a distributed instructional environment and proposes a conceptual model. We conclude by discussing a design approach for a prototype integrated instructional system that utilizes the model in the domain of battleship management.

2.0 Networks and Expert Systems

Networking is an important tool in the integration of not only hardware but also software. Networks enable users of local processors to employ a rich array of useful resources in different locations including programs and data (Martin, 81). A series of networked systems can also make distance seem transparent. Thus, the user need not even know which node/s hold which software and the overall system responses can be coordinated automatically.

Another area from which this research draws upon is Expert Systems. By representing facts and heuristics in a knowledge base, it is possible to build systems that execute in a non-procedural manner (Barr and Feigenbaum, 1982). Since these systems have an inference engine, they can be made to adapt intelligently to the learner's strengths, weaknesses, priorities and style of learning (Carbonell, 1970; Clancey, 1985). This helps to avoid rambling through irrelevant pre-determined steps depending on the problem at hand as well as streamline the dialogue and keep it directly focussed on the real issues. In fact, this has been the major objective of Computer-Aided Diagnostic and Prescriptive Instruction (CADPI).

The advantage of networking CADPI systems can be summarized as follows.

1. The proximity requirement between the location of the instructional system and the location of the student can be removed.

2. The student need not know which instructional system exists in which node. Based on the level of knowledge, style and needs of the learner, the network can automatically determine which system is best suited to coach a particular student in a certain topic.

3. Depending on the complexity of the topic and the problems solved, geographically dispersed instructional systems in similar domains can consult each other and coordinate their operation in a participative style.

4. Instructional systems in complementing domains can also function cooperatively.

3.0 A Conceptual Model

Figure 1 depicts a conceptual model of the roles, interactions, and geographical dependencies involved in a distributed instructional environment. It must be noted that the model is system independent in that the model would hold irrespective of the number of instructional systems and where they are located in the network as also the number of domains of knowledge whether they be overlapping or complementing. However, once the level of the interactions among the model components are known, the model does have implications for the eventual physical design.

The model identifies three roles in a typical instructional setting. They are the i) Super-Ordinate (SP), ii) Role-Incumbent (RI), and iii) Sub-Ordinate (SB). The Super-Ordinate represents the role of an ideal instructor, one who has reached the limits of perfection in teaching in a specific domain. It connotes both maturity and experience in the level of knowledge and teaching effectiveness. The Role-Incumbent may be considered as the role played by a less mature and inexperienced instructor. In this sense, the Role-Incumbent is partially a student himself in that he/she is yet to perfect his/her knowledge and teaching skills. Thus, a Role-Incumbent may require assistance in organizing the course content, the style and strategies to be used in presenting the various topics, setting up assignments, tests and the like. The Sub-Ordinate captures the role of a student who is trying to assimilate the knowledge in a certain course for the very first time.

The three roles lead to three major distinct interactions,

1. Super-Ordinate   Role-Incumbent

   Here the RI is trying to perfect his knowledge and teaching skills.

2. Role-Incumbent   Sub-Ordinate

   This is the typical instructional environment. The student is tutored by a semi-skilled teacher.

3. Super-Ordinate   Sub-Ordinate

This is the ideal situation where the student is fortuante to have the best teacher in a particular topic in the whole country.

It must be noted that there are several other interactions as well that have been captured in the model. For example, multiple Super-Ordinates either in the same domain or complementing domains may need to interact in certain complex tutoring situations. Similarly, the Role-Incumbents may also wish to interact among themselves as colleagues.

From the foregoing model, it is possible to imagine how an instructional system involving multiple roles in distributed environment will actually function. Each role may be considered as representing a Knowledge Base at a particular node in the overall system network. It is easy to see that a high level of participation and coordination among the various nodes are absolutely essential for successful system performance.

4.0 Design Issues

The above discussion leads us to an array of difficult design issues that need to be resolved before a network-based integrated system can be implemented. The major ones among them are, (i) the distribution of knowledge among the various nodes in the network, (ii) the design of the core expert system, and (iii) the design of the network topology.

4.1 Distribution of Knowledge

The distribution of knowledge is an important consideration that directly impinges both on the efficiency and effectiveness of any distributed system. There are three different strategies that can be used in the distribution. These are,

Centralized Knowledge Base:

Under this strategy, all the knowledge is pooled into a central knowledge base located in a single host computer. This provides maximum control of the network, reduces problems of integration arising from incompatibility among different machines, and allows employment of powerful software-languages, expert and database systems and other useful application systems in the manipulation of the knowledge base. It has the further advantage that multiple copies of the various subsets of the knowledge base need not be kept at different locations. Redundancy and contradictions among the rules stored in the knowledge base can also be easier identified and quicker resolved if the entire knowledge is stationed in one location. However, the centralization strategy generally involves much higher communication costs since the central knowledge base has to be accessed always for each consulting session from the user locations.

Hierarchical Knowledge Bases:

In this strategy, knowledge can be layered into various level of abstraction from the highest level meta-knowledge about the various aspects of the knowledge domain to the lowest level problem-specific knowledge. Each layer is then allocated to a node in the network. The knowledge base can also be split in terms of their geographic characteristics, and type of usage. In the hierarchical set up, the different knowledge bases may have a certain degree of dependence on each other. A special component is needed at the top of the hierarchy that carries the meta-knowledge to be able to interleave the rest of the knowledge bases in its command in appropriate patterns in a time and problem dependent fashion.

Separated Knowledge Bases:

A third strategy is to employ separated knowledge bases in the network. Here each knowledge base, although part of a single overall network, is inherently independent. The advantage of this approach is that there is considerable autonomy among the different knowledge bases and each can solve most of the problems that occur in the instructional sessions in their own locations without having to access the core-expert in the network. This configuration is isomorphic to real-world instruction in that typically each instructor and his students function as an independent group in its respective location without regard to what goes on external to the group in other locations.

4.2 Design of the Core Expert

The most challenging part of designing an integrated network of knowledge bases is the design of its meta-level knowledge base. Architecturally, we refer to this knowledge base as the Core Expert. Issues such as removal of redundant knowledge and resolution of contradictory knowledge at independent locations can best be settled through the incorporation of a core expert accessible from any location in the network. Figure 2 illustrates the various categories of meta-level expertise the core expert should have to be able to coordinate an instructional session that involves concurrently the interactions among the three roles captured in our conceptual model, viz., the Super-Ordinate, the Role-Incumbent, and the Sub-Ordinate. First, the core expert should hold the meta-knowledge about the types of knowledge domains represented in the network. In other words, the network system should have the self-knowledge as to what kinds of subjects, say mathematics, history, geography, that it is qualified to tutor. Second, the core expert must have the ability to know how many Super-Ordinate and Role-Incumbent knowlege bases are connected to the network and where they are located so that user queries arising during a session can be directed to the appropriate node. Third, the core expert must also have some abstract knowledge of the structural characteristics of each knowledge domain so that it can decide which Super-Ordinate/Role-Incumbent knowledge bases are to be roll-called to participate in a particular session. Finally, the core expert module also have the inferential and control knowledge to direct/coordinate the session dynamically during its run.

4.3 Design of Network Topology

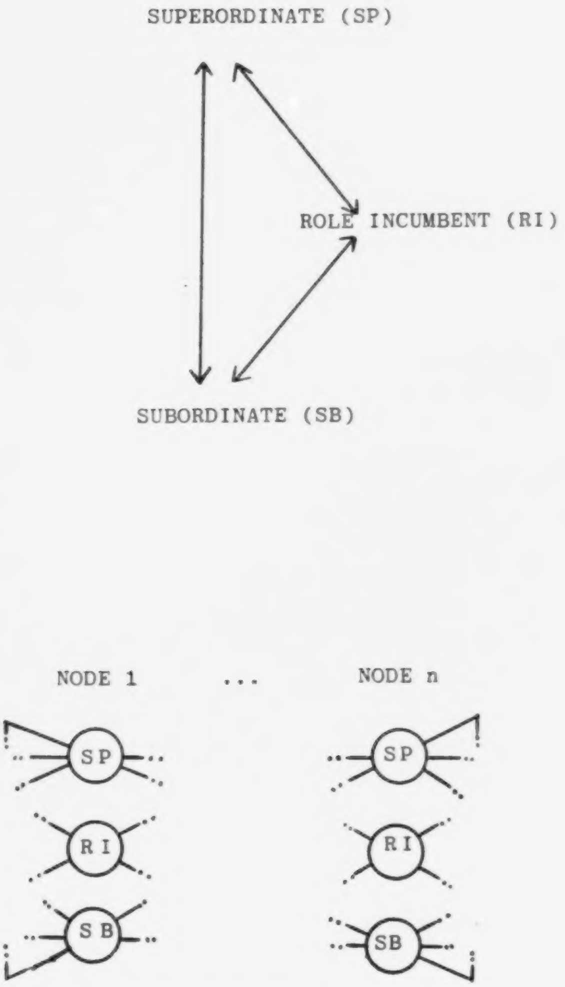
The term topology, in the context of communications network, refers to the way in which the nodes (or the knowledge bases in our case) are connected to one another. It determines the data paths that may be used between any pair of nodes. In general, topology influences the performance, reliability, and control strategies implemented in the network.

We shall consider four topologies that are relevant to our discussion. They are, the star network, the hierarchical ring. The star network consists of a set of user terminals that directly interconnect with a central system where all the knowledge bases are stored. In this case, two users can interact only through the central facility. This topology is most suitable when the knowledge distribution strategy is the centralized one. The hierarchical network, on the other hand, is a centralized knowledge base in star configuration with a set of nodes that are themselves centers of other stars. Where the knowledge distribution strategy employed is hierarchical, this topology is perfect. In the case of ring networks, each knowledge base is connected to its neighboring knowledge base in equal relationship. Each knowledge base can independently conduct tutoring sessions without having to access a central knowledge base so long as the required level knowledge is available in the local knowledge base. This topology is ideal for the separated knowledge base strategy we discussed earlier.

We propose a hybrid configuration, the hierarchical ring, for the operationalization of our conceptual model. Since the expert module plays the coordinating role among the various knowledge bases and since these knowledge bases are potentially independent, such a configuration will be very efficient (Figure 3).

5.0 Prototype Implementation

To demonstrate the feasibility of the model, a prototype distributed system, called the Tactical Action Officer Instructional System (TADIS) is being implemented (Zivovic 1986). The system provides instruction to officers in the domain of tactical decision making during a simulated naval battle. The motivation for this domain came about from the Strategic Computing Initiative promulgated by the Defense Advanced Research Project Agency (DARPA). The prototype is implemented in EXSYS shell and runs on a microcomputer. It utilizes production rules for knowledge representation and backward chaining technique for inferencing. The current version of the system has 63 rules in its knowledge base. A network of microcomputers is being acquired after which the multiple knowledge bases representing the various instructional roles will be coded. Figure 4 shows the Process Model used in the TADIS. A sample run of the system is shown in Figure 5.


SUPERORDINATE (SP)
ROLE INCUMBENT (RI)
SUBORDINATE (SB)
NODE 1
...
NODE n
SP
RI
SB
SP
RI
SB

Superordinate == Expert / Specialist

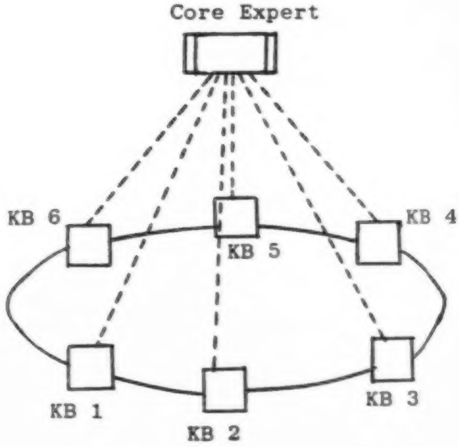
Role Incumbent == Instructor / Teacher

Subordinate == Learner / Student

Figure 1 Conceptual Model of the Roles, Interactions, and Their Geographic Dependencies

Knowledge on the types of domains that can be instructed through the network of tutoring systems (Example domains: Mathematics, History, Geography)
Knowledge on how many Superordinate and Role Incumbent knowledge bases are in the network and where they are located
Knowledge on the structural characteristics on each domain so that Core Expert can decide which set of systems in the network are to be accessed during an instructional session
Control and inferential knowledge to conduct a session and dynamically manipulate all the accessible knowledge bases depending on actual needs during run time

Figure 2 Meta-Level Knowledge in the Core Expert



KB = Knowledge Base

Figure 3 A Hierarchical-Ring Configuration

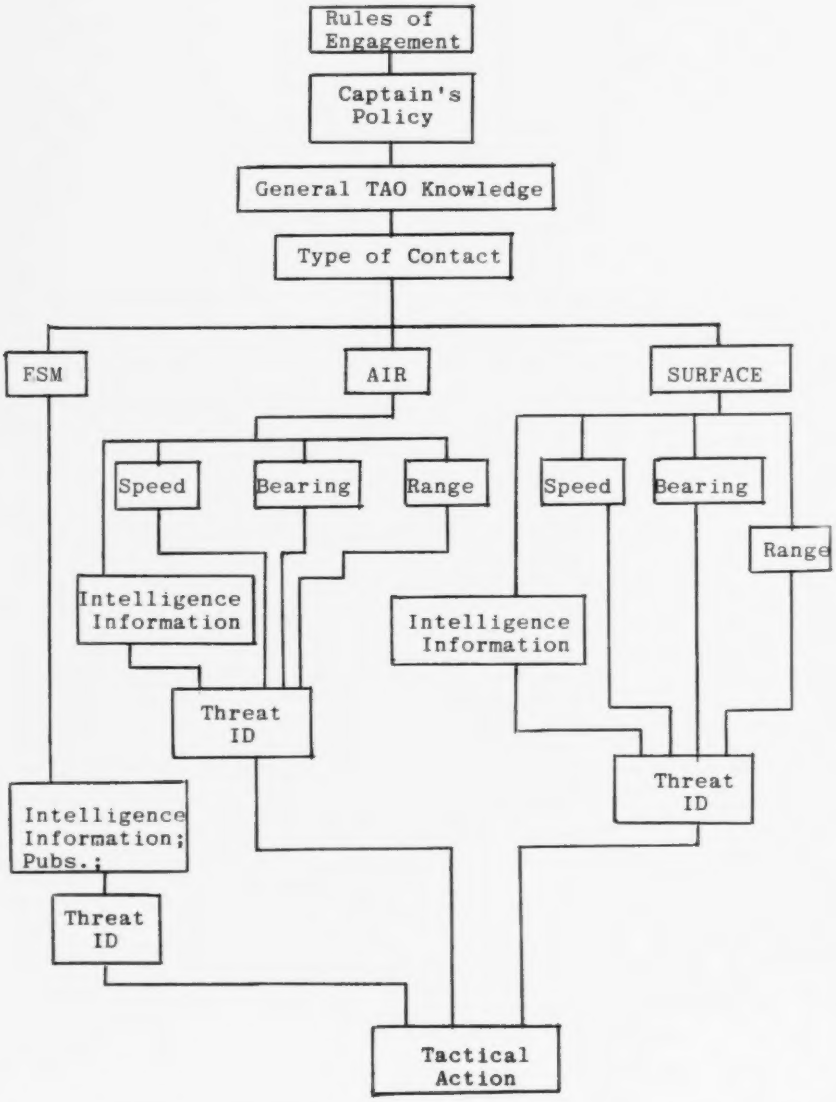

Rules of Engagement
Captain's Policy
General TAO Knowledge
Type of Contact
FSM
AIR
SURFACE
Speed
Bearing
Range
Speed
Bearing
Range
Intelligence Information
Intelligence Information
Threat ID
Threat ID
Intelligence Information; Pubs.;
Threat ID
Tactical Action

Figure 4 Process Model in the Tactical Action Officer Instructional System (TAOIS)

```
type of contact is
      1     air contact
      2     surface contact
      3     ESM contact

            1,3

you hold
      1     no surface contact
      2     no air contact
      3     surface contact
      4     air contact on 2D radar
      5     air contact on 3D radar

            4

you have
      1     no visual ID
      2     visual ID

            1

visibility is
      1     good
      2     not good

            1

air contact speed is
      1     very fast
      2     fast
      3     slow
      4     unknown

            2

you detect
      1     MissileHoming ESM
      2     Targeting A/C ESM
      3     Missile A/C ESM
      4     Gunship ESM
      5     Missileship ESM
      6     No ESM
      7     Enemy ESM

            3

5 consecutive bearings are
      1     the same
      2     not the same

            1
```

```
each consecutive range is
     1   smaller
     2   same
     3   larger
     4   unknown

         1

These are recommendations only.
TAO is still required to make
the final decision.

     1    engage with terrier     :  10/10
     2    continue to monitor its
          position and/or ESM     :  10/10
     3    engage with 5/54        :   5/10
     4    alter your course and
          speed to bring the
          enemy inside the range
          of your weapon(s)       :   5/10
     5    alter your course and
          speed to move out of the
          range of enemy's weapon(s):5/10
```

Figure 5 A Sample Instructional Session with the TAOIS

References

(Barr and Feigenbaum, 1982)
A. Barr and E.A. Feigenbaum (Eds), The Handbook of Artificial Intelligence, HeurisTech Press, 1982.

(Carbonell, 1970)
J.R. Carbonell, AI in CAI: An Artificial Intelligence Approach to Computer Assisted Instruction, IEEE Transactions on Man-Machine Systems, Vol. MMS-11, No. 4, 1970.

(Clancey, 1985)
W.J. Clancey, Acquiring, Representing, and Evaluating a Competence Model of Diagnostic Strategy, Report No. STANS-CS-85-1067, Stanford University Stanford, August, 1985.

(Martin, 1981)
J. Martin, Design and Strategy for Distributed Data Processing, Prentice-Hall, 1981.

(Simon, 1973)
H.A. Simon, The Structure of Ill-Structured Problems, Artificial Intelligence: 181-201, 1973.

(Sivasankaran and Ruby, 1986)
T.R. Sivasankaran and R. Ruby, Jr., A Design-Theoretic Approach for a Network-Based Intelligent Instructional System, Conf. on Technology in Training and Education, Montgomery, March 10-13, 1986.

(Zivovic, 1986)
S. Zivovic, Can an Expert System Help Train Tactical Action Officers: Some Experience from an Early Prototype, Naval Postgraduate School, Monterey, March, 1986.

Knowledge on the types of domains that can be instructed through the network of tutoring systems

(Example domains: Mathematics, History, Geography)

Knowledge on how many Super-Ordinate and Role-Incumbent knowledge bases are in the network and where they are located

Knowledge on the structural characteristics on each domain so that Core Expert can decide which set of systems in the network are to be accessed during an instructional session

Control and Inferential knowledge to conduct a session and dynamically manipulate all the accessible knowledge bases depending on actual needs during run time

Figure 2 Meta-Level Knowledge in the Core Expert

```
type of contact is
   1   air contact
   2   surface contact
   3   ESM contact

       1,3

you hold
   1   no surface contact
   2   no air contact
   3   surface contact
   4   air contact on 2D radar
   5   air contact on 3D radar

       4

you have
   1   no visual ID
   2   visual ID
       1

visibility is
   1   good
   2   not good
       1
air contact speed is
   1   very fast
   2   fast
   3   slow
   4   unknown
       2

you detect
   1   MissileHoming ESM
   2   Targeting A/C ESM
```

3 Missile A/C ESM
4 Gunship ESM
5 Missileship ESM
6 No ESM
7 Enemy ESM

3

5 consecutive bearings are
1 the same
2 not the same

1

each consecutive range is
1 smaller
2 same
3 larger
4 unknown

1

These are recommendations only. TAO is still required to make the final decision.

1 engage with terrier : 10/10
2 continue to monitor its position and/or ESM : 10/10
3 engage with 5/54 : 5/10
4 alter your course and speed to bring the enemy inside the range of your weapon(s) : 5/10
5 alter your course and speed to move out of the range of enemy's weapon(s): 5/10

Figure 5 A sample Instructional Session with the TADIS

NOTE: CAP/PI is a total computer assisted diagnostic and prescription instruction package. CAD/PI evaluates all data input by the student and prescribes addition work based on student weaknesses. All quizzes, tests, and "outside assignments" are generated by the computer. In additon, all work is graded directly from the student disk to a teacher management disk (electronic grade book) using defaults (correct responses, time limits, numerical equivalents for alphabetic grades, etc.) established by the instructor. CAD/PI allows the instructor to be a true manager of the learning environment. MICRONUMERICS (1985) and WORD PROCESSING FOR MICROCOMPUTER (1986) were co-authored by Dr. Ruby and are working examples of the CAD/PI approach. Both textbooks are published by Science Research Associates (SRA).

Contact: Ralph Ruby, Jr.
Arkansas State University
Box 2771
State University, AR 72467

501-972-3040

COGNITIVE ERROR DIAGNOSIS AND PRESCRIPTION FOR FRACTION ARITHMETIC

Doris Smith Shaw, Ph.D.
Associate Investigator, Research
U.S. Army Corps of Engineers
Construction Engineering Research Laboratory

Abstract

Much of the research in computerized instructional design has focused upon aspects of the instruction itself. While such studies are important to improve the effectiveness of computer-based education, attention needs to centered upon characteristics of the learner. In this study, it was found that brief interactive dialogue could increase scores of students who were diagnosed, using the Rule-space approach, as having a particular misconception. The instruction was not as worthwhile for students whose needs were different.

Introduction

Much of the research concerning computerized instructional design has focused upon particular aspects of the instruction itself. Concerns with objectives, learning strategies, graphics, fantasy, question types, drill and practice algorithms, knowledge of results, lesson management, user "friendliness", and tutorial feedback are some of the areas that have been studied.

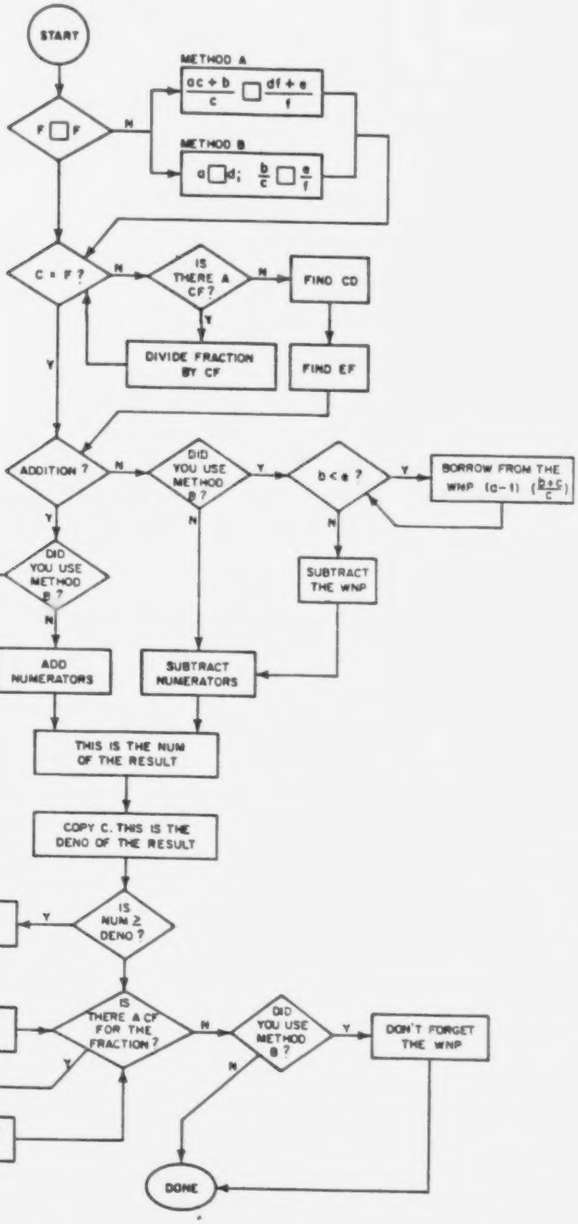
While results of these investigations are important to improve the effectiveness of computer-based education, more attention needs to be centered upon characteristics of the learner. In this study a diagnosis of the developmental level of the student with respect to a conceptual task structure of the subject domain, fraction arithmetic, was the first step. Then the effects of two types of computerized remediation, both aimed at diagnosed misconceptions, were evaluated.

Method

Jr. high school students were given an adaptive pretest that diagnosed borrowing error patterns in fraction subtraction. All subjects then randomly received one of three treatments: either of two types of computerized borrowing-specific remediation, or no treatment at all. The two remediation programs were: (1) a concept based dialogue, and (2) a practice lesson on a series of borrowing items with knowledge of results and correct answers provided. After six weeks, the same subjects were posttested and their scores and diagnosed misconceptions were compared with the pretest.

The diagnosis was based upon an analysis of student responses to test items, determined from a procedural network (Klein, Birenbaum, Standiford, & Tatsuoka, 1981). These items were then further investigated through protocol studies (Shaw, Standiford, Klein, & Tatsuoka, 1982). A task specification chart (Figure 1) for the subject domain was proposed (Birenbaum & Shaw, 1985) which specified the structure of conceptual development in the subject domain.

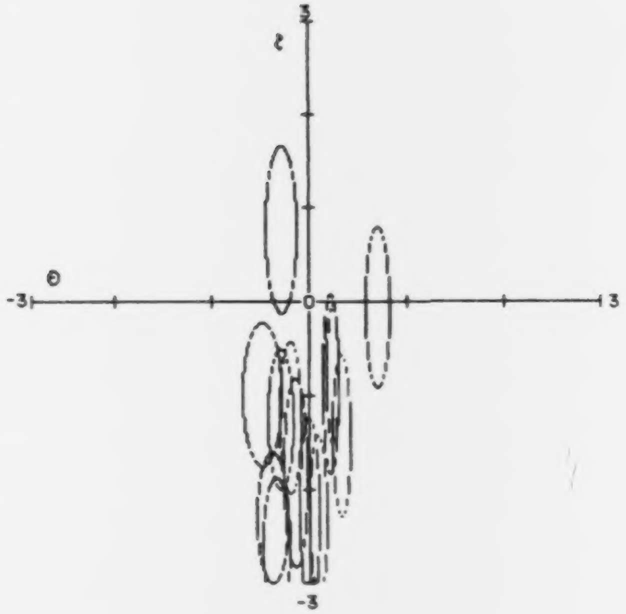
The computerized adaptive test was designed by K.K. Tatsuoka and Baillie on the University of Illinois PLATO system. It was developed utilizing test responses on the above test items from 595 students to establish parameters of difficulty and discrimination according to item response theory (IRT). These characteristics and information about deviation from "normal" performance on the set of items predicted the misconception areas in the Rule-space (Tatsuoka, 1983) as shown in Figure 2. An item tree (Figure 3) was constructed such that the chains of items would reveal the student's conceptual status according to his method of work as shown in Figure 1.


START
F □ F
N
METHOD A
(ac + b)/c □ (df + e)/f
METHOD B
a □ d; b/c □ e/f
C = F ?
N
IS THERE A CF?
N
FIND CD
Y
DIVIDE FRACTION BY CF
FIND EF
Y
ADDITION ?
N
DID YOU USE METHOD B ?
Y
b < e ?
Y
BORROW FROM THE WNP (a−1) ((b+c)/c)
N
N
Y
ADD THE WNP
DID YOU USE METHOD B ?
N
SUBTRACT THE WNP
ADD NUMERATORS
SUBTRACT NUMERATORS
THIS IS THE NUM OF THE RESULT
COPY C. THIS IS THE DENO OF THE RESULT
DIVIDE NUM BY DENO
Y
IS NUM ≥ DENO ?
DON'T FORGET THE WNP
IS THERE A CF FOR THE FRACTION ?
N
DID YOU USE METHOD B ?
Y
DON'T FORGET THE WNP
Y
N
DIVIDE FRACTION BY CF
DONE

$a\frac{b}{c} \pm d\frac{e}{f}$;

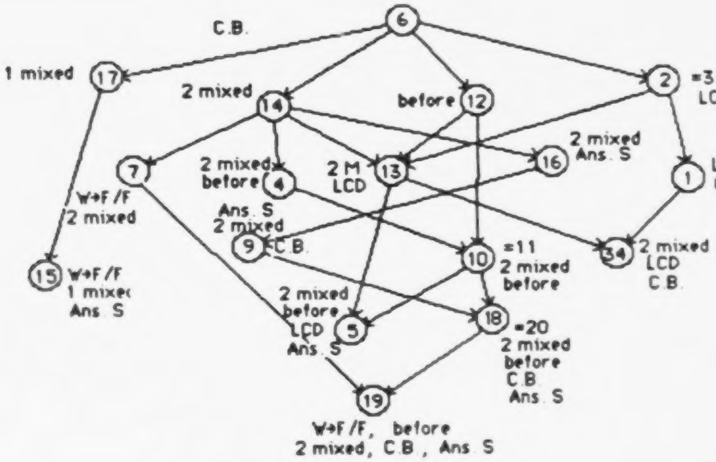
F = fraction;
CD = common denominator;
CF = common factor;
WNP = whole number part;
NUM = numerator; DENO = denominator; EF = equivalent fraction.

Figure 1. Task Specification chart.

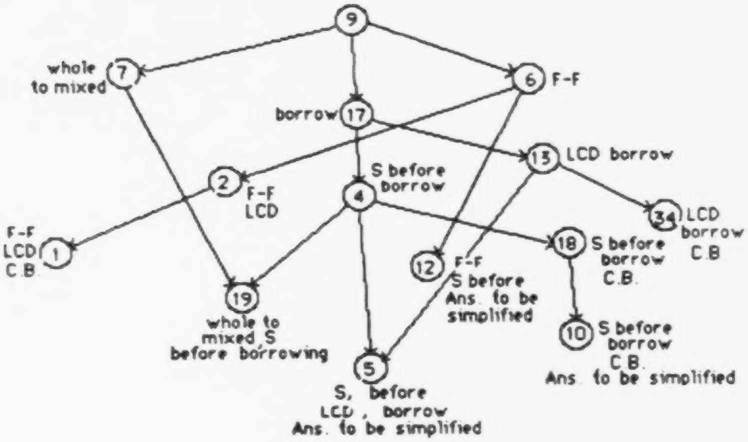


Student responses tend to cluster around the centroid representing each rule. The area within an ellipse represents perfect and imperfict applications of the rule.

Figure 2. Ellipses showing the areas of the borrowing errors in rule-space.


6
C.B.
1 mixed 17
2 mixed 14
before 12
2 =3 LCD
7
2 mixed before 4
2 M LCD 13
16 2 mixed Ans. S
1 LCD C.B.
W+F/F 2 mixed
Ans. S 2 mixed 9 C.B.
10 =11 2 mixed before
34 2 mixed LCD C.B.
15 W+F/F 1 mixed Ans. S
2 mixed before LCD 5 Ans. S
18 =20 2 mixed before C.B. Ans. S
19
W+F/F, before 2 mixed, C.B., Ans. S

Item Tree by Method A


9
whole to mixed 7
6 F-F
borrow 17
13 LCD borrow
S before borrow
2 F-F LCD
4
34 LCD borrow C.B
F-F LCD 1 C.B.
18 S before borrow C.B.
12 F-F S before Ans. to be simplified
19 whole to mixed S before borrowing
10 S before borrow C.B. Ans. to be simplified
5 S, before LCD, borrow Ans. to be simplified

Item Tree by Method B

Item characteristics are listed next to the item number.

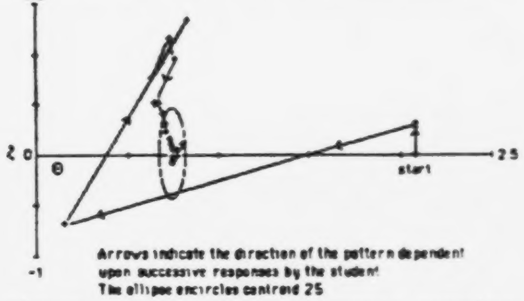
F= Fraction; LCD = Least Common Denominator; CB = column borrow

Figure 3. Item trees for fraction subtraction test.

Rationale

Information from wrong responses was studied in an attempt to discover misconception areas, much as in the "Impasse and Repair" theory proposed by VanLehn (1982). For example, if a student converted 3 4/12 to 3 5/12 in order to subtract from the mixed number, it was predicted that he had subtracted 1 from the whole number part and added 1 to the numerator of the fraction part. If he converted 3 4/12 to 3 14/12, it seemed probable that he had subtracted 1 from the whole number part and added 1 to the 10's column of the numerator of the fraction part, much as in column subtraction borrowing. In both cases it seemed that he understood the additive property of the fraction and the whole number in a mixed number, he could subtract 1 from the whole number, but he was unable to add 1 to the fraction part correctly. When those students were asked, "How many 12ths are there in 1?" their response was erroneous. This led to the conclusion that the two "bugs", or incorrect repairs, resulted from the same impasse, failure to understand the conceptual identity, "1 = x/x". Both would be classified as "borrowing misconceptions" for the purpose of diagnosis.

A related problem was seen in the case of students who incorrectly responded that 2 - 1/3 = 2 1/3. They had difficulty with the concept that 2 = 1 3/3 and it might be predicted that their problem would stem from the same identity impasse as the above erroneous responses. However, it might be that their misconception only dealt with the "renaming" of whole numbers. The adaptive testing routine needed to provide a method for making such distinctions. An example of the items and responses in an actual case is seen in Table 1. Figure 4 shows the mapping of successive responses which diagnosed the student in ellipse 25, or the area which indicates that the student has a misconception dealing with borrowing from whole numbers. Refer to Figure 3 for the item sequence on the item tree.



Arrows indicate the direction of the pattern dependent upon successive responses by the student
The ellipse encircles centroid 25

Figure 4. Rule-space response mapping.

The first correct response is mapped at 2.08,0. When the student enters a wrong response on item four, the direction changes leading to the response mapped at 15,-69. As the items are answered, each new point is calculated. When the responses are sufficiently close together, the testing is concluded. In this case the last eleven responses fall within the ellipse for group 25.

It was necessary for the computer program to store successive steps in the student's problem solving, similar to "showing work" on paper, in order to differentiate between responses. A student might convert a mixed number to an improper fraction and make a column subtraction error which would give the same wrong answer as a different "bug" resulting from a borrowing misconception. Students were encouraged work at the computer rather than on scratch paper in order to discover such differences.

Results

It was found that a brief interactive dialogue was capable of increasing the scores of students who were diagnosed as having the misconception. The instruction was not as worthwhile for students whose needs were different. In addition, characteristic shifts in the Rule-space were observed for students in the different groups. See Figure 5 for an explanation of the rule-space mapping and Figure 6 for the plot of the shifts for the first experimental group. Table 1. reports the t-test results for comparison of pretest and posttest in each group.

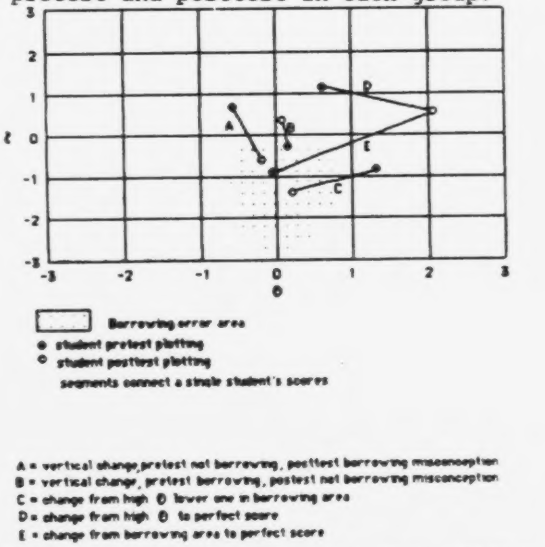
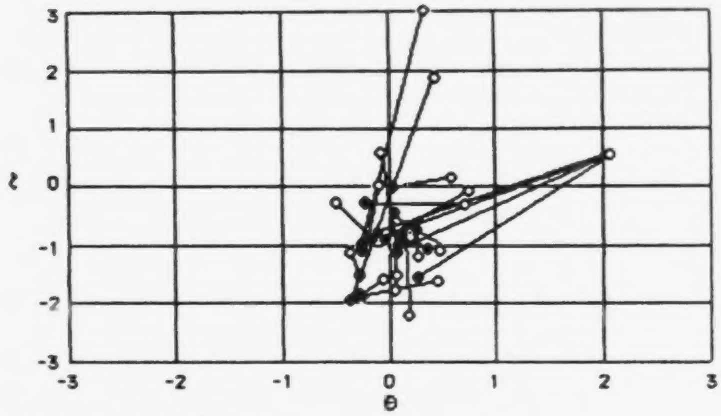


Figure 5. Examples of rule-space shifts.



● student pretest plotting
○ student posttest plotting
segments connect a single student's scores

Student pretest and posttest data points for Group 1

ID	θ_1	ζ_1	θ_2	ζ_2	ID	θ_1	ζ_1	θ_2	ζ_2
106	-0.25	-1.89	-0.04	-1.61	402	0.26	-0.73	0.20	-2.23
112	-0.02	-0.92	0.75	-0.07	404	-0.15	-0.91	-0.49	-0.28
114	-0.25	-1.00	0.35	4.33	807	0.10	-1.01	2.08	0.55
116	0.26	-0.78	0.29	-1.19	808	-0.26	-1.86	0.46	-1.62
128	0.08	-1.54	0.08	-1.54	903	-0.24	-1.09	-0.08	0.02
209	0.06	-0.44	0.48	-1.10	905	0.28	-1.55	2.08	0.55
210	-0.21	-0.27	0.73	-0.30	914	-0.36	-1.95	0.44	1.87
211	-0.10	-0.82	2.08	0.55	1003	0.05	-0.01	0.60	0.15
304	0.08	-1.14	-0.07	0.58	1005	0.14	-0.88	2.08	0.55
310	-0.26	-1.54	-0.35	-1.15	1013	0.38	-1.08	0.21	-0.88
318	0.15	-0.78	0.06	-1.81					

Figure 6. Group 1: Borrowing errors with remedial dialogue treatment.

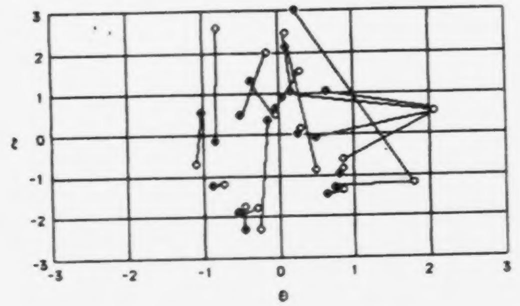


Figure 7. Group 2: Not borrowing errors with remedial dialogue treatment.

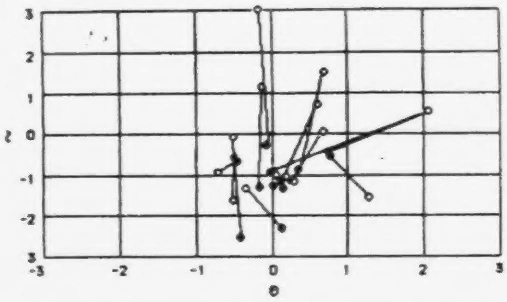


Figure 8. Group 3: Borrowing errors with practice (KOR) treatment.

Table 1.

Correlation Tables, Groups 1-6

	θ1	θ2	θ diff
θ1		0.2694	0.0020
θ2	0.2694		* 0.9636
θ diff	0.0020	* 0.9636	

GROUP 1 (n = 21)

	θ1	θ2	θ diff
θ1		* 0.7299	-0.0839
θ2	* 0.7299		* 0.6199
θ diff	-0.0839	* 0.6199	

GROUP 2 (n = 21)

	θ1	θ2	θ diff
θ1		* 0.7451	0.4276
θ2	* 0.7451		* 0.9215
θ diff	0.4276	* 0.9215	

GROUP 3 (n = 15)

	θ1	θ2	θ diff
θ1		* 0.8785	-0.3804
θ2	* 0.8785		0.1076
θ diff	-0.3804	0.1076	

GROUP 4 (n = 17)

	θ1	θ2	θ diff
θ1		* 0.5674	0.1675
θ2	* 0.5674		* 0.9069
θ diff	0.1675	* 0.9069	

GROUP 5 (n = 20)

	θ1	θ2	θ diff
θ1		* 0.7797	0.0859
θ2	* 0.7797		* 0.6908
θ diff	0.0859	* 0.6908	

GROUP 6 (n = 16)

CORRELATIONS BETWEEN STUDENT THETA VALUES

θ1 = pretest
θ2 = posttest
θdiff = θ2 - θ1
* indicates significance at the .05 level

In summary, the only group showing a significant change in pretest and posttest scores was Group 1, the group receiving misconception-specific dialogue treatment. Students in Group 2 who did not make borrowing errors tended to find the correct rule only when they already had high pretest scores, and Groups 3 and 4 who received the practice treatment for their problems moved away from their past pattern, but developed other "bugs". Of special interest were the control groups who received no remedial treatment. Group 5, with borrowing "bugs", exhibited a random pattern, some moving to other misconceptions and others not. Group 6 tended to develop borrowing error patterns between the two testing periods. These results can be observed quite easily from the rule-space representation and have important implications for instruction.

Discussion

The mathematical model, Rule-space, makes it possible to gereralize the adaptive testing procedure as well as map the outcome of instruction. The development of appropriate test items is basic for two reasons. First, the item parameters are a requirement for the determination of the true score (the "x" axis in the rule-space concept), and the items serve as a tool for making observations and structuring the item tree.

For example, present work in teaching the use of automated graphics programs for architectural design has the same kind of item requirements even though the subject domain is different. Questions concerning the processes of representation of space and line in a computer must be developed in order to measure the user's conceptual level, in much the same way that questions were developed in fraction arithmentic. Studies of wrong approaches or inefficient methods of work can lead to a skills heirarchy or task structure. Preliminary protocol studies reveal differences in mental models for specific classes of users. The use of "shortcuts" seems to be characteristic of "experts" or those who really understand, as observations of borrowing procedures often revealed. All subject domains have similar attributes which must be discovered.

Of particular value in the Rule-space concept is the mapping of changes in order to describe and evaluate the effects of instruction. Without graphic representation, it would be difficult to draw conclusions concerning the kinds of conceptual shifts occurring with different instructional strategy, which is essential to remediation.

Conclusion

In conclusion, the results of this study support use of diagnostic test results and a conceptual approach to remediation. Testing is needed to learn enough about the student to provide effective remediation. Misconception-based remediation does seem to produce results and does not lead to development of other bugs, which we saw in the groups not receiving treatment and which has been shown in other studies of procedure-based remediation (Davis & McKnight, 1980; Resnick, 1982). Computerized adaptive testing made misconception-specific remediation not only possible but practicable. Computers can thus link diagnosis and prescription in instruction.

References

Birenbaum,M., & Shaw, D. (1985). Task specification chart: A key to a better understanding of test results. Journal of Educational Measurement. 22, 219-230.

Davis, R.B., & McKnight,C. (1980). The influence pf semantic content on algorithmic behavior. Journal of Mathematical Behavior, 3(1), 39-87.

Klein, M.F., Birenbaum, M., Standiford, S.N., & Tatsuoka, K.K. (1981). Logical error analysis and construction of tests to diagnose student "bugs" in addition and subtraction of fractions. Urbana, Il:University of Illinois, CERL.

Resnick, L.B. (1982). Syntax and semantics in learning to subtract. In T.P.Carpenter, K.M. Moser, & T.A. Romberg(Eds.), Addition and subtraction, a cognitive perspective. Hillsdale:Lawrence Erlbaum Associates.

Shaw, D. (1986). Effects of adaptive diagnostic testing on two types of computerized remediation. Unpublished dissertation. Urbana, IL: University of Illinois.

Shaw, D., Standiford, S., Klein,M., & Tatsuoka, K.K. (1982). Error analysis of fraction arithmetic - selected case studies. (Research Report 82-2-NIE). Urbana, IL: University of Illinois.

Tatsuoka, K.K. (1983). Rule space: An approach for dealing with misconceptions based on item response theory. Journal of Educational Measurement, 20(4), 345-354.

VanLehn, K. (1982). Bugs are not enough: Empirical studies of bugs, impasses and repairs in procedural skills. Journal of Mathematical Behavior, 3(2), 3-50.

ADCIS Proceedings, November 1986

A COMPUTER-BASED KNOWLEDGE EXTRACTION TOOL:

A STEP IN THE DEVELOPMENT OF A COGNITIVE SKILLS TUTOR

Mary L. Stoddard and Joe D. Emerson, Los Alamos National Laboratory

Richard Kern, U.S. Army Research Institute

ABSTRACT

Los Alamos National Laboratory, under the sponsorship of the U.S. Army Research Institute, is developing an experimental computer-tutor to be used as part of the Armor Officer's Basic Course at Fort Knox, Kentucky. The tutor's objective is to train students to apply the types of cognitive processing strategies needed to more effectively organize their knowledge for application in planning and conducting tactical operations. The tutor is being developed through an iterative process, with the first phase being a knowledge extraction computer-based tool. Student knowledge organization in this domain will be obtained through collection of on-line and off-line performance data. The tool is designed to obtain the knowledge organization through a motivating, realistic tactical operations exercise.

INTRODUCTION

Process of Knowledge Extraction

The need for tools to facilitate the process of knowledge extraction is now widely recognized (e.g. Feigenbaum and McCorduck, 1983; Chovan, 1986). The high degree of efficiency afforded to the MYCIN expert system by the TEIRESIAS knowledge extraction tool is the primary example of such a tool (Davis, 1982), but development of such tools is difficult and seemingly domain specific. Decision support tools (e.g. Expert Choice) can be considered for use as knowledge extraction tools, although they are not specifically designed for this purpose. In addition, motivating subjects to use such tools is often a challenge as the tools tend to be unentertaining and the results of efforts not obvious. Knowledge extraction conducted by knowledge engineering interviews is a tedious, time-consuming effort (e.g. Madni, Freedy, and Ahlers, 1985; Griffin, 1985) and is often a bottleneck in many artificial intelligence projects. The knowledge extraction process is an essential first step toward the creation of intelligent tutoring systems and could be facilitated by the availability of motivating, computer-based tools to supplement interviewing. Knowledge extraction is now receiving attention from researchers in the computer-based training field (e.g. Hall and Karim, 1986; Lasnick, 1986; Roberts, 1986).

Context of This Project

The Training Research Team, Military Systems Group, Los Alamos National Laboratory, is developing an experimental computer-tutor under the sponsorship of and in cooperation with the U.S. Army Research Institute (ARI). Dr. Richard Kern, ARI's contracting officer representative in 1985, performed an analysis of student performance on tactical operations in the Armor Officer Basic Course at Fort Knox, Kentucky. The course consists of 928 hours of instruction presented in 15 weeks, with the mastery tests being conducted both in the field and in the classroom on terrain boards that are used to simulate field situations. Kern discovered that student-accumulated knowledge from the first 10 weeks of instruction was not organized to relate to performance of tactical operations and that students tended to persist in using simple cognitive skills (e.g. rote memory). Success in a tactical operations exercise depends upon knowledge of previously learned facts and procedures and ability to relate and prioritize among these facts and procedures in varying situations. Consequently, many students are not able to anticipate what they might do and why or how and when they should execute a particular type of tactical operation under varying enemy and terrain conditions.

Knowledge Organization and This Project

Research comparing "experts" with "novices" has highlighted the importance of determining the extent and organization of an individual's knowledge of a particular domain (e.g. Greeno, 1980; Simon, 1980; Norman, 1980). More recently, research on representation of student knowledge structures in a particular domain has moved into the education and training arena (e.g. Diekhoff, 1983; Eylon and Reif, 1984; Fredericksen, 1984). These researchers have constructed models that represent knowledge structures reflected in the content of instruction and in the instructor's or other expert's knowledge of the domain. They have used a variety of techniques to represent students' knowledge structures before and after the instruction. These researchers typically report that the effects of the instruction are reflected in the development of students' knowledge structures. Thus, this strand of research lends credence to the notion that techniques for representing organization of a person's knowledge of a domain can be used to monitor effectiveness of training.

The objective of the computer-tutor being developed is to improve student officers' ability in planning, decision-making and executing tactical operations. To accomplish this objective, programs will be designed to cause the student to apply cognitive processing strategies considered effective in organizing the knowledge worked within each program. In addition, techniques will be incorporated into these programs for capturing and representing organization of the student's knowledge. Thus, these representations will provide a way of evaluating the effectiveness of the cognitive strategies used as well as the effectiveness of knowledge organization development on external performance.

The approach is two phased. Phase I is pure research, where a computer-based interactive video exercise in tactical operations is used to extract student knowledge organization and to compare instructor (theoretical expert) performance and thinking with that of students. Phase II is a development phase in which the tutorial portion will be built. The expert knowledge organization will be used as a template to determine appropriate feedback. The tutor will diagnose each student on the basis of student performance in the same type of exercise as that used in Phase I and prescribe appropriate tutorials.

Focus of This Paper

The remainder of this paper focuses on the Phase I prototype for student knowledge extraction. First, a description of the prototype design is presented. This is followed by a description of the data collection and analysis methodology. In conclusion, some discussion of the strengths and weaknesses of this approach and the potential for the use of this approach in the development of intelligent tutoring systems in other domains is provided.

DESIGN OF THE KNOWLEDGE EXTRACTION PROTOTYPE TOOL

Project Phases Described

The Phase I goal is to discover how student armor officers organize their knowledge regarding tactical operations. Rather than perform live knowledge engineering-type interviews with students, a computer-based tool is being used to facilitate the extraction process. Students are not inherently interested in providing their knowledge organization; thus, a motivating tool is needed. Level III interactive video is used to increase the degree of realism. A tactical operations exercise was selected as a vehicle for knowledge extraction because humans naturally like problem solving and a relevant task-oriented exercise would be more motivating than an interview format would be. The Phase I prototype not only provides cues for knowledge organization but also provides a test-bed for the initial Phase II computer-tutor design. In the Phase I prototype, student performance is collected and judged only in a few places where judgement is necessary in order to proceed through the simulation.

Phase I Prototype Objectives

The Phase I prototype is divided into two parts: planning and execution. The objective of the planning part is that: given the elements of a tank company operations order to conduct a deliberate attack, the armor officer would be able to develop a scheme of maneuver for a tank platoon attack plan and to explain the rationale for the plan. The objective of the execution part is that during the execution of the planned scheme of maneuver, the armor officer would be able to direct changes in maneuver based on scenario changes (i.e. enemy contact, obstacles, or order modification). In the field, the armor officer takes notes when an operations order is given to him, and in the same way, the prototype gives the user an opportunity to highlight the parts of the order that he deems to be significant.

User Interface

The hardware user interface of the prototype consists of a microprocessor-based workstation with interactive video disk. A video disk produced for another course at Fort Knox is being reprocessed for this effort. User input is accomplished through the use of a mouse and a keyboard. Digital audio, as well as video-disk audio is used. The program contains an advanced organizer showing a video portion of an actual exercise, students inputting a route and formations on notional terrain, computer graphics showing views of changes in situation (e.g. encountering a steep bank or artillery fire). The Pilot Plus authoring language is being used. A conceptual diagram of the program is shown in Figure 1.

Student Scenario

How is the prototype experienced by the student user? The following is a brief example of one pathway through the program that will give the reader a feeling for a typical session.

> A student approaches the terminal and sees an initial screen with a title "Combat Unit Leader Trainer." He clicks the mouse once to start the program and sees a brief video/audio sequence of a tactical operation. This is the "advanced organizer": a scene to give him a feeling for what to expect and to motivate him to get involved. The screen then fades to an armor captain who tells him to stand by for an operations order. He is prompted to give his name and is then provided with a few instructions about use of the hardware and screen options. There is a command options bar on the bottom of each screen enabling him to exit, go forward, go back, or obtain help. He is then presented with the operations order by the captain and is allowed to save portions of the order in an electronic notebook for retrieval later. Then he is asked what actions he wants to take in light of the given situation. He inputs the route, techniques, and formations that he thinks are appropriate. He is prompted to provide verbally his rationale for each of his decisions; the verbal response is recorded on an audiotape. This concludes the planning part of the program.
>
> The student then proceeds into a segment of the disk where he watches as the mission he has planned is executed. As the simulation proceeds, new situations are introduced (e.g. a creek too deep to ford, sniper fire on a nearby hill, instructions to stop and await further orders). The student must then input his actions, and again, he is prompted to provide the rationale for his actions. At these critical points the consequences of his actions will be shown visually. The session concludes with the student either achieving the mission or failing to achieve the mission.

DATA COLLECTION AND ANALYSIS METHODOLOGY

Eight sources of data, both on line and off line, will be used to help build a representation of each student's knowledge organization. The on-line script will be annotated after the experiment using the off-line data so that a full case study of each student may be compiled. Patterns of student knowledge organization will then be sought from examination of the scripts. These patterns will form the basis for the Phase-II tutor, facilitating the provision of highly individualized feedback and learning prescriptions.

On-Line Data Sources

1. Source: Phrases from operations order highlighted by the student.
   Use: Highlighted information provides clues as to what information the student considers necessary for task completion.

2. Source: Student's maneuver plan: route, techniques, formation.
   Use: Clues to student's planning skills and synthesis of recalled facts and procedures.

3. Source: Time to complete task.

Use: Trace the time required for each stage of the process; calculate ratio of think time to action time for each student; relate time to degree of success.

4. Source: Student's performance in execution when situation changes.
   Use: Clues to student's decision making and synthesis of recalled facts and procedures.

5. Source: Access of help/aids/electronic notebook.
   Use: For learning which clues the student thinks that he needs to successfully complete the exercise. To learn what type of information is most frequently in need of retrieval.

Off-line Data

6. Source: Handwritten notes.
   Use: Clues to students mental schema of the problem.

7. Source: Student's rationale for plans and moves.
   Use: Provides information on student reasoning behind decisions.

8. Source: Pre- and post-interviews.
   Use: Provides background information to use in extracting categories; enables experimenter to identify possibly significant variables.

DETERMINING PATTERNS IN STUDENT KNOWLEDGE ORGANIZATION

The case studies will be analyzed to

- compare student's solution with the "school solution,"
- observe deviations of the student's solution from the school solution,
- determine the student's rationale for his solution,
- identify variables that might be influencing student deviation from the school solution, and
- seek patterns in the deviations.

The precise method of identifying patterns of problem-solving behavior will be determined after the data have been compiled. Selection of a method has been deferred because the optimal analysis method will not become apparent until the data are examined. We have, however, considered several possible approaches, which may be used in combination:

- Use an existing error analysis taxonomy (e.g. Norman, 1982). Norman developed a classification of "slips" based on their presumed sources. The three main classes of slips are slips in the formation of intention, slips resulting from faulty activation and slips in triggering of schema.
- Use the mental model approach (Gentner and Stevens, 1982). A mental model is a person's view of the task he is asked to perform. A mental model contrasts with a conceptual model, which is the view of the task invented by the instructor or doctrinal experts and is useful as a teaching tool (the "school solution"). Identifying deviations between the two models provides specific information to guide instructional design.
- Use the Component Display Theory (CDT) of Merrill (1983) which provides a performance taxonomy, based upon memory structures, that is designed to be used in an early stage of instructional design.
- Use a thinking style taxonomy (Harrison and Bramson, 1982). Give students a pretest on their thinking style and sort their solutions to the problem according to their thinking style. Determine if there is a correlation.
- Develop an error classification system tailored specifically for the tactical operations domain.

IMPLICATIONS OF THIS APPROACH FOR DEVELOPMENT IN OTHER DOMAINS

Strengths of This Approach

The cognitive/development approach described here is advantageous for a project such as this for the following reasons:

- o It is a motivating way to extract knowledge organization from students.

- It is readily compatible with other off-line methods of knowledge extraction.
- It provides specific information useful for designing highly individualized computer-based training.
- The prototype serves as a test-bed for the computer-tutor.
- The knowledge engineer's role is facilitated.
- The realistic exercise allows early involvement of the user and facilitates implementation in the school.

Weaknesses of This Approach

This cognitive/development approach does, however, have some weaknesses:

- The amount of data to be compiled and analyzed is large.
- The tool for knowledge extraction provides clues to knowledge organization, but it risks not providing enough clues for the researcher to fully understand student problem-solving behavior in this domain--hence the dependence on off-line data collection methods.
- An active, full-time observer is necessary.
- The student may be frustrated at the lack of tutorial information as part of the exercise.

The development team, in a tradeoff analysis process, concluded that for this particular application the strengths outweigh the weaknesses. The degree to which this approach is transportable to other knowledge domains and other research-/development projects depends upon such a tradeoff analysis. For example, if the computer-based product being developed does not need to be highly sensitive to individual differences and if it is needed quickly, the performance of the knowledge extraction phase described here may not be cost effective.

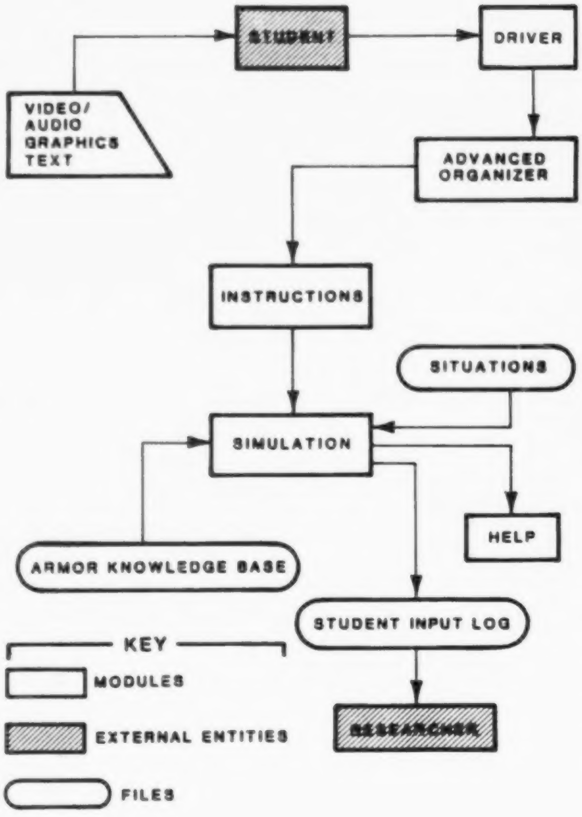

STUDENT
DRIVER
VIDEO/
AUDIO
GRAPHICS
TEXT
ADVANCED ORGANIZER
INSTRUCTIONS
SITUATIONS
SIMULATION
HELP
ARMOR KNOWLEDGE BASE
STUDENT INPUT LOG
KEY
MODULES
EXTERNAL ENTITIES
FILES
RESEARCHER

Figure 1

REFERENCES

Chovan, J. "Prototype Tools for Information Extraction to Be Used in the Creation of Computer-based Training Materials." Paper presented at the 1986 ADCIS Conference, New Orleans, Louisiana.

Davis, R. Teiresias. "Applications of Meta-Level Knowledge." In Knowledge-Based Systems in Artificial Intelligence, R. Davis and D. Lenat, Eds. (McGraw-Hill, New York, 1982).

Diekhoff, G. "Testing through Relationship Judgements." Journal of Educational Psychology 75(2) (1984): 227-233.

Eylon, B. and F. Reif. "Effects of Knowledge Organization on Task Performance." Cognition and Instruction 1(1) (1984): 5-44.

Feigenbaum, E. and P. McCorduck. The Fifth Generation. (Addison-Wesley, Menlo Park, California, 1983).

Fredericksen, N. "Implications of Cognitive Theory for Instruction in Problem Solving." Review of Educational Research 54(3) (1984): 363-407.

Gentner, D. and A. Stevens. Mental Models. (Lawrence Erlbaum Associates, New Jersey, 1983).

Greeno, J. "Trends in the Theory of Knowledge for Problem Solving." In Problem Solving and Education: Issues in Teaching and Research, D. Tuma and F. Reif, Eds. (Lawrence Erlbaum Associates, New Jersey, 1980).

Griffin, A. "Methodologies for Extracting Knowledge: Building an Expert System for Training Simulators." Proceedings from the 7th Interservice/Industry Training Equipment Conference, Orlando, Florida, October 1985.

Hall, K. and M. Karim. "Business Simulation Program Documentation." Paper presented at the 1986 ADCIS Conference, New Orleans, Louisiana.

Harrison, A. and R. Bramson. Styles of Thinking. (Anchor Press, New York, 1982).

Lasnick, V. "Theoretical Perspectives of Knowledge Representation and Acquisition in the Development of Computer-based Instructional Systems." Paper presented at the 1986 ADICS Conferencc, New Orleans, Louisiana.

Madni, A., A. Freedy, and R. Ahlers. "An Expert Surrogate Instructor for Acquisition of Cognitive and Motor Skills in Rifle Marksmanship." Proceedings from the 7th Interservice/Industry Training Equipment Conference, 1985.

Merrill, D. "Component Display Theory." In Instructional Design, Theories, and Models: An Overview of Their Current Status, C. Reigeluth, Ed. (Lawrence Erlbaum Associates, New Jersey, 1983).

Norman, D. "Cognitive Engineering and Education." In Problem Solving and Education: Issues in Teaching and Research, D. Tuma and F. Reif, Eds. (Lawrence Erlbaum Associates, New Jersey, 1980).

Norman, D. "Steps Toward a Cognitive Engineering: Design Rules Based on Analysis of Human Error." Proceedings of the Conference on Human Factors in Computer Systems, Gaithersburg, Maryland, 1982.

Roberts, F. "Extending Knowledge Acquisition Aids for Instruction." Paper presented at the 1986 ADCIS Conference, New Orleans, Louisiana.

Simon, H. "Problem Solving and Education." In Problem Solving and Education: Issues in Teaching and Research. D. Tuma and F. Reif, Eds. (Lawrence Erlbaum Associates, New Jersey, 1980).

ADCIS Proceedings, November 1986

A COGNITIVE ERROR DIAGNOSTIC ADAPTIVE TESTING SYSTEM

UNIVERSITY OF ILLINOIS

KIKUMI TATSUOKA

ABSTRACT

This paper introduces a new probabilistic approach for diagnosing cognitive errors by selecting an optimal set of items tailored to an individual examinee. The method is based on Item Response Theory and developped to a general theory so that it is applicable to any domain of interest. However, it will involve a painstaking task analysis.

INTRODUCTION

Finding the sources of misconceptions possessed by students is a difficult task because it is impossible to see what is happening in their brains. The only observable outcomes are the students' responses on the test items. Studying their spoken-aloud protocols is one method for discovering how students solve or think through problems. During the past decade cognitive scientists have employed various Artificial Intelligence techniques to develop systems for diagnosing students' misconceptions (Brown & Burton, 1978). These methods infer unobservable cognitive processes from logical relationships among cognitive tasks involved in a domain of interest.

A new approach toward attaining the same goal, called Rule Space, has emerged recently in Psychometrics. The model has been developed by Tatsuoka and her associates (Tatsuoka, 1983; Tatsuoka & Tatsuoka, in press) and has been successfully applied to arithmetic domains such as signed-numbers, fraction addition and subtraction, and matrix algebra.

RULE SPACE MODEL

The model has been developed to solve a specific classification problem—one in which the entities to be classified are what are called rules of operations that one can use insolving a problem in some well-defined domain such as arithmetic, algebra and the like. A correct rule, by definition, produce the right answer to all the items in a test, but sometimes wrong rules may produce the right answer to some subset of items. Moreover, some rules are the combinations of two or more different wrong rules. If we construct the test items carefully so that the predicted erroneous rules can be expressed uniquely as binary item response patterns specified in a subtask tree, or process network (Marshall, 1981; Tatsuoka,1984), then the rules can be distinguished by examining the binary response pateerens, or vectors.

Now, two kinds of information are included in a response vectors; how many ones (quantitative), and which items are ones (qualitative). The former represents the levels of achievement, θ while the latter is characterized by a new variable, ζ introduced by Tatsuoka (1984b), which is an index expressing atypicality of response pattern. The response pattern produced by a rule in the "bug library" is expressed by an ordered pair of θ and ζ. If the rule is atypical for the high ability students, then the rule point will be located in a rightupper part of the space. If the rule is typical for lower ability levels, then the rule point will be in a left-lower part of the rule space.

The values of θ and ζ are determined by characteristics of the testitems which are expressed by probability functions in Item Response Theory. A basic assumption underlying IRT is that the probability of a response can be determined by an examinee's ability level and the characteristic of each item such as difficulty, and discriminating power (Lord, 1980).

SEMANTIC MEANING OF IRT CURVES

The probability functions (IRT curves) are determined by the values of item parameters. Therefore, it is important to investigate how the values of the parameters are influenced by the nature and characteristics of subtasks defined in a task analysis. Since it is impossible to discuss generally as to how curves are affected by underlying cognitive tasks, the illustration of phenomenon will be given here in the context of fraction addition domain. The procedure being described below will be, however, applicable to any general domain.

Two methods of adding two fractions, a b/c + d e/f are considered for carrying out a logical task analysis. Method A is thatmixed numbers are always converted to simple fractions of the b/c + e/f type and then the regular procedure is followed. But Method B is observed in more advanced students than Method A is. They separate the whole number parts from the fraction parts and add them separetely. Adding the two fraction parts will be just the procedure of adding two simple fractions. These subtasks are analyzed carefullyand represented in a process network as is shown in Figure 1. At each node of the process tree, a circled number represents a subtask described inside a diamomd or a rectangle. The numbers written near rectangles are codes of bugs originating from the subtask described in the rectangles. Thus, the number 1 means a student must distinguish the item type of a problem as either F+F (addition of two simple fractions) or M+M (addition of two mixed numbers). The number 23 means the examinee must make equivalent fractions after obtaining the lowest common multiple of two different denominators. There are four different types of denominators. The product of c and f is the least common denominator , 11, one is a multiple of the other 12 and 13 and c and f have a common factor, 14. Klein et al (1981) predicted 80 different rules of operation from 23 different subtasks.

As can be seen in Figure 1, the relationships of 23 subtasks are expressed by a graph of directed arcs. The 36 items are constructed so that every subtasks or some combination of subtasks are involved in some items. For instance , in order to get the right answer by method A, the item 3/5 + 1/5 involves Subtasks 1,7,10,15,16, 18, and 19, while 1/2 + 1 10/7 requires Subtasks 1,2,6,7,10,11,23,15,16,17,18, and 19. Item 1/5 + 2 5/3 requires the same subtasks as 1/2 + 1 10/7 does. Then, their item response curves will have very similar curves as shown in Figure 2.

CLASSIFICATION OF RESPONSE PATTERNS

Tatsuoka and Tatsuoka (in press) introduced the notion of a "bug distribution" where the centroid of the distribution corresponds to the perfect application of the rule. Alist of the rules in the bug library can be transformed into the list of the centroids and covariance matrices of multivariate normal distributions. If all erroneous rules (and the correct one) were to be mapped into the rule space along with their neighboring response patterns representing random slips from them, the resulting topography would be something like what is seen in Figure 3. That is, the population of points would exhibit modal densities at many rule points that each forms the center of an envelopping ellipse with the density of points getting rarer as we depart farther from the center in any direction. We may assert that the set of ellipses gives a complete characterization of the rule space. By this is meant that, once these ellipses are given, any response-pattern point can be classified as most likely being a random slip from one or another of the erroneous rules (or the correct one). We have only to determine, for a suitable percent value, which one of the several ellipses uniquely includes the given point.

Various pattern classification techniques such as classification by Mahalanobis generalized-squared distance of an examinee's position from the centroids associated with the rules stored in the bug library, or Bayes'decision rules for minimum errors can be used to diagnose misconceptions (Tatsuoka,1985).

ADAPTIVE TESTING

In the context of the rule space, Tatsuoka, Tatsuoka and Baillie (1983) have developed an adaptive testing procedure by which a student's rule can be diagnosed. A computer program that will select a set of optimal items for each individual and administer it, and then terminate the testing as soon as a certain stopping rule is satisfied has been written on the PLATO system. The strategy of selecting a sequence of items is an extension of the method commonly used in IRT - based adaptive testing procedure. By so doing, the accuracy of the estimated values of and will be attaiend most quickly. The procedure is an application of the theory of convergence in the functional space— a set of item response functions. The last figure is an example of adaptive testing result in fraction addition problems. Since the example is very unusual rule, the centroid point associated with the rule lies a upper part of the space, (-.15,5.54).

The numbers near "+" signs are the order of performances, 4 is the point corresponding to the response pattern of 4 items, 5 is that of five items, previous 4 items and a new item.

As can be seen in Figure4, 21 responses to them is very close to the cluster around the rule.

The first column is the number of the items already given, the second column shows the binary scores of 36 items and the third shows actual item numbers. and values are given in 4th a and 5th columns, and the distance of the current point from the centroid of the rule is given in the last column. A necessarly information to diagnose this student's performance on the 36-item test is already obtained from the first 21 items and, hence, testing can be stopped there.

Brown, J. S., & Burton, R. R. (1978). Diagnostic models forprocedural bugs in basic mathematical skills. Cognitive Science, 2, 155-192.

Klein, M., Birenbaum, M., Stadiford, S., & Tatsuoka, K. K. (1981). Logical error analysis and construction of tests to diagnose student "bugs" in addition and subtraction of fractions. (Research report 81-6-NIE) Urbana, ILL.: University of Illinois, CERL

Marshall, S. P. (1980). Procedural networks and production systems in adaptive diagnosis. Instructional Science, 9, 129-143.

Tatsuoka, K. K. (1983). Rule space: An approach for dealing with misconceptions based on item response theory. Journal of Educational Measurement, 20,4, 345-354.

Tatsuoka,K. K. (1984). Analysis of erros in fraction addition and subtraction problems. (Final report for Grant NIE-G-82-0002). Urbana, Ill.: University of Illinois, CERL.

Tatsuoka,K. K.(1984b). Caution indices based on item response theory. Psychometrika, 49,1, 95-110.

Tatsuoka,K.K., Tatsuoka,M.M. & Baillie, R. (1984). Application of adaptive testing to a fraction test. (Research report 84-3-NIE) Urbana, Ill.: University of Illinois, CERL.

Tatsuoka,K.K. (1985). A probabilistic model for diagnosing misconceptions in the pattern classification approach. Journal of Educational Statistics, 12,1,55-73.

Tatsuoka, K.K.& Tatsuoka, M.M. (in press). Bug distribution and pattern classification. Psychometrika.

Lord, F. M. (1980). Application of item response theory to practical testing problems. Hillsdale,NJ: Lawrence Erlbaum.

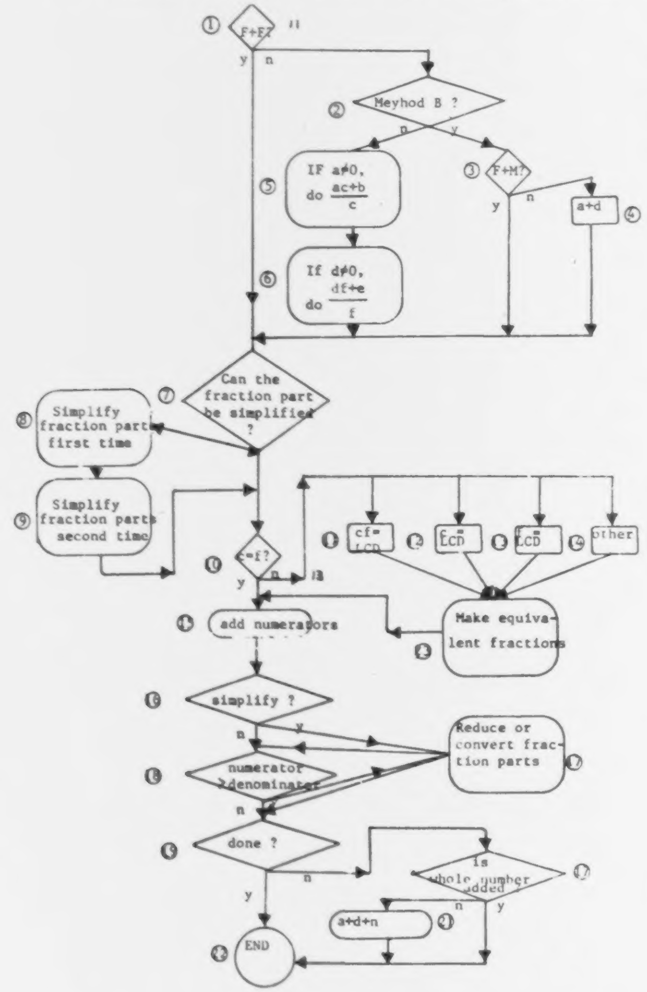

①
F+E?
y
n
②
Meyhod B ?
n
y
⑤
IF a≠0, do ac+b / c
⑥
If d≠0, do df+e / f
③
F+M?
y
n
a+d
④
⑦
Can the fraction part be simplified ?
⑧
Simplify fraction part first time
⑨
Simplify fraction part second time
⑩
c=f?
y
n
cf= LCD
Ecd
LCD
other
add numerators
Make equivalent fractions
simplify ?
n
y
Reduce or convert fraction parts
numerator ≥denominator
n
done ?
y
n
is whole number added
n
y
a+d+n
END

Figure 1 A subtask tree (process network) of fraction addition problems

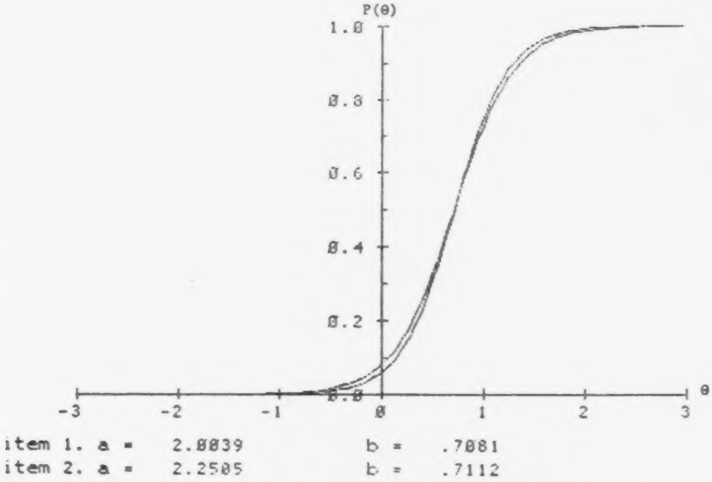


Figure 2 The item response curves of two items underlying cognitive tasks are identical

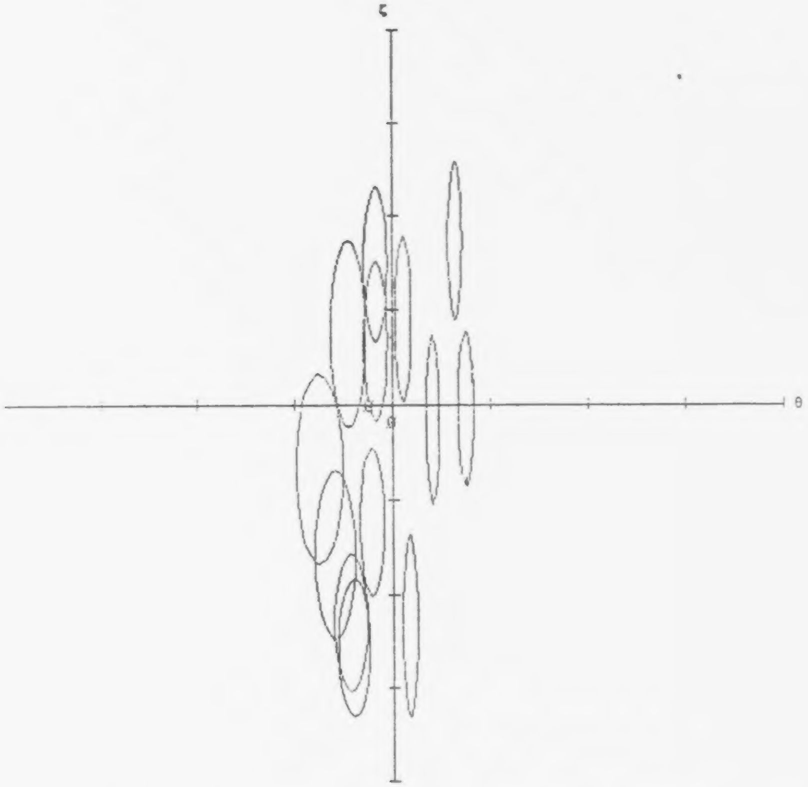


Figure 3 Fifteen Ellipses Representing Fifteen Error Types Randomly Chosen From Forty Sets of Ellipses

ndone	score	next item	θ	ζ zeta	I(θ)	rdist
4:	0	21	0.32	-0.04	4.21	5.60
5:	0	4	0.13	0.46	5.74	5.09
6:	0	14	0.10	0.31	5.82	5.24
7:	1	1	0.18	-0.09	8.05	5.65
8:	0	30	0.16	-0.26	7.99	5.81
9:	0	7	0.13	-0.35	8.11	5.90
10:	1	35	0.18	-0.63	9.90	6.18
11:	1	32	0.25	-0.40	11.18	5.95
12:	1	24	0.29	-0.59	12.75	6.15
13:	0	15	0.23	-0.05	13.20	5.61
14:	0	33	0.18	0.68	13.22	4.88
15:	0	25	0.12	1.43	13.06	4.12
16:	1	34	0.15	1.09	15.58	4.46
17:	0	9	0.10	2.04	15.73	3.51
18:	0	27	0.07	2.92	15.71	2.63
19:	0	6	0.04	3.82	15.63	1.73
20:	0	16	0.01	3.97	16.21	1.58
21:	0	11	-0.06	5.29	15.64	0.26
22:	0	2	-0.07	5.41	15.66	0.15
23:	0	13	-0.09	5.47	15.88	0.09
24:	0	12	-0.11	5.54	15.83	0.04
25:	0	23	-0.13	5.55	16.10	0.03
26:	0	36	-0.14	5.48	16.22	0.06
27:	0	10	-0.15	5.51	16.22	0.04
28:	1	3	-0.10	5.56	17.95	0.05
29:	0	17	-0.12	5.55	18.33	0.03
30:	1	18	-0.08	5.51	20.08	0.08
31:	0	20	-0.09	5.54	20.23	0.06
32:	0	29	-0.11	5.56	20.35	0.05
33:	0	19	-0.12	5.57	20.48	0.04
34:	0	5	-0.13	5.56	20.61	0.03
35:	0	8	-0.14	5.55	20.64	0.01
36:	0	**	-0.15	5.54	20.60	0.00

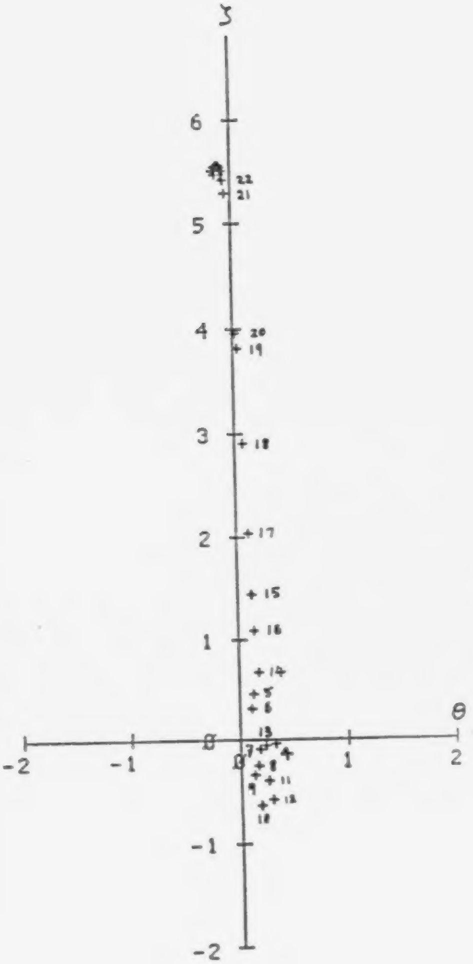


Figure 4 Simulation of adaptive testing algorithm, Bug = Group2 in fraction addition problems : First 4 item scores = 0100.

An Application of Rule Space Methodology to the Assessment of Tonal Memory

Walter Vispoel

University of Illinois at Urbana Champaign

Abstract

This paper describes the application of rule space error diagnostic procedures to a newly developed computer-administered adaptive test of memory for melodies. In addition to the typically-provided ability score, this test provides a listing of specific types of melodies that an examinee is good or poor at remembering. Seventy five percent of the error patterns of the 163 examinees in the norming sample were successfully diagnosed using this procedure. Highly reliable ability scores and accurate error classifications were obtained using an average of 41 items. Error classification frequencies for subsamples of graduate music students, college undergraduates and high school students are provided and compared.

Introduction

Commercially available standardized tests of musical ability share certain features and limitations. They are all paper and pencil tests in which each examinee responds to the same set of items. Typically, raw scores and norms are the descriptive information one obtains. Although these data are useful for assessing one's relative standing on a trait of interest, they provide limited information about an examinee's knowledge base. These tests fail to address the question of why an examinee answers certain items correctly and others incorrectly. Rule space methodology was developed to answer this question efficiently (Tatsuoka, 1983, 1985). This paper describes its use in a newly developed computer-administered test of tonal memory.

Test Development

Every major musical ability test battery includes a test of memory for melodies (e.g., Bentley, 1966; Drake, 1957; Gordon, 1966, 1979: Seashore, Lewis & Saetveit, 1960; Wing, 1960). The item format used in the present test is similar to the one used in the Seashore, Wing and Bentley tonal memory tests. A typical test item requires an examinee to compare two versions of a melody. First, a melody is played. Then, the melody is replayed but one note is altered. Respondents are asked to specify the number of the note that has been changed.

Two hundred and seventy eight such items were developed. Item difficulty indices (i.e., "b" values) based on the item response theory two-parameter logistic model were calibrated on a sample of 467 high school and college students. To construct item categories for the subsequent use of rule space procedures, it was necessary to determine which characteristics of the test melodies had the most impact on item difficulty. These melodies differed on several dimensions including length (i.e., the number of notes, ranging from 4 to 9), tonality (i.e., tonal versus atonal), extent of the note alteration (i.e., number of half steps), position of the altered note (i.e., beginning, middle or end), direction of the alteration (i.e., up or down) and range (i.e., the number of half steps between the lowest and highest note in a melody).

The "b" or item difficulty parameters were regressed on these item characteristics using a forward stepwise selection procedure. Results indicated that length and tonality were the best predictors of item difficulty (see Vispoel, 1986 for further details). Specifically, two relationships emerged. First, as melody length increased, memory decreased. Second, atonal melodies were harder to remember than tonal melodies. These relationships formed the basis for two assumptions subsequently used to construct item categories. The assumptions are: If an examinee cannot remember a tonal or atonal melody of a given length, then he or she will not be able to remember a longer melody of the same tonality; and if an examinee cannot remember tonal melodies of a fixed length, he or she will not be able to remember atonal melodies of the same length. The 27 resulting item groups are described in Table 1.

Results

A subset of 93 representative items from the original 278-item pool was administered to a sample of 163 respondents including 46 high school students, 85 college undergraduates and 32 graduate music students. As Table 2 shows, 75% of these students were successfully classified into one of the 27 groups. Membership in a particular group describes the types of melodies that a given examinee can and cannot remember. For example, if an examinee is classified into group 21, he or she tends to remember all melodies except atonal melodies longer than 6 notes. Test output provides both an ability score and a listing of melody types that an examinee is good or poor at remembering. Percentages of high school students, college undergraduates and graduate music students classified into each group and their mean ability scores are given in Table 2. Note that classification results were best for music students (i.e., 87.5% were successfully classified). Of those classified, 82% were good at remembering tonal melodies of any length. These results contrast sharply with those for successfully classified high school students and college undergraduates, who remembered tonal melodies of any length only 11.4 and 16.9% of the time, respectively. Music students, it appears, have more highly developed schemas for traditionally-structured music than do non-music majors.

The examinees' responses were then used to simulate their performance on an adaptive test using subsets of the preceding 93 items. The simulated adaptive test was terminated after scoring an average of 41 items per examinee. The ability scores derived from the stimulated adaptive test and the original 93 item test had a correlation of .98. There was also a 99% agreement in error group classifications between the two procedures. These results demonstrate the impressive efficiency and precision of rule space adaptive testing procedures. A version of this test has been implemented on the PLATO computer system.

Conclusion

The advantage of rule space methodology, then, is that it makes efficient concurrent assessment of an examinee's ability and knowledge base. This information describes why an examinee answers certain items correctly and others incorrectly. In the present case, correct and incorrect responses are determined by tonality and length of the melodies. After completing an adaptive test using rule space methodology, an examinee is given a norm-referenced tonal memory score and information about specific types of melodies he or she is good or poor at remembering.

References

Bentley, A. (1966). Musical ability in children and its measurement. London: Harrap.

Drake, R. K. (1957). Manual for the Drake Musical Aptitude Tests. Chicago: Science Research Associates.

Gordon, E. (1965). Musical aptitude profile manual. Boston: Houghton Mifflin.

Gordon, E. (1979). Primary measures of music audiation: Test manual. Chicago: G.I.A.

Seashore, C.E., Lewis, P., & Saetveit, J. C. (1960). Manual of instructions and interpretations for the Seashore Measure of Musical Talents (2nd revision). New York: The Psychological Corp.

Tatsuoka, K. K. (1983). Rule space: An approach for dealing with misconceptions based on item response theory. Journal of Educational Measurement, 20, 345-353.

Tatsuoka, K. K. (1985). A probabilistic model for diagnosing misconceptions by the pattern classification approach. Journal of Educational Statistics, 10 (1), 55-73.

Vispoel, W. P. (1986). *An adaptive test of musical memory: An application of item response theory to the assessment of musical ability*. Unpublished Doctoral Dissertation, University of Illinois at Urbana-Champaign.

Wing, H. D. (1960). *Manual for standardized tests of musical intelligence*. Windsor: Nat. Gound. Educ. Res. Publ.

Table 1

Item Groups Indicating Melody Types that Examinees Can and Cannot Remember

Item Group Number	Melody Types[a] Can Remember	Cannot Remember
1	4T	5-9T, 4-9A
2	4-5T	6-9T, 4-9A
3	4-6T	7-9T, 4-9A
4	4-7T	8-9T, 4-9A
5	4-8T	9T, 4-9A
6	4-9T	4-9A
7	4T, 4A	5-9T, 5-9A
8	4-5T, 4A	6-9T, 5-9A
9	4-6T, 4A	7-9T, 5-9A
10	4-7T, 4A	8-9T, 5-9A
11	4-8T, 4A	9T, 5-9A
12	4-9T, 4A	5-9A
13	4-5T, 4-5A	6-9T, 6-9A
14	4-6T, 4-5A	7-9T, 6-9A
15	4-7T, 4-5A	8-9T, 6-9A
16	4-8T, 4-5A	9T, 6-9A
17	4-9T, 4-5A	6-9A
18	4-6T, 4-6A	7-9T, 7-9A
19	4-7T, 4-6A	8-9T, 7-9A
20	4-8T, 4-6A	9T, 7-9A
21	4-9T, 4-6A	7-9A
22	4-7T, 4-7A	8-9T, 8-9A
23	4-8T, 4-7A	9T, 8-9A
24	4-9T, 4-7A	8-9A
25	4-8T, 4-8A	9T, 9A
26	4-9T, 4-8A	9A
27	4-9T, 4-9A	

[a] T = Tonal, A = Atonal, "-" = through

Table 2

Frequencies and Percentages of Examinees in Each Item Group Within the Total Sample and Graduate Music Student, High School Student and College Undergraduate Subsamples

Item Group Number	Melody Types[a] Remembered	Graduate Music Students Freq.	%	High School Students Freq.	%	College Under-Graduates Freq.	%	Total Sample Freq.	%
1	4T								
2	4-5T			1	2.2	3	3.5	4	2.5
3	4-6T			1	2.2	4	4.7	5	3.1
4	4-7T			2	4.3	5	5.9	7	4.3
5	4-8T					2	2.4	2	1.2
6	4-9T	2	6.3			2	2.4	4	2.5
7	4T, 4A			2	4.3			2	1.2
8	4-5T, 4A			3	6.5	3	3.5	6	3.7
9	4-6T, 4A			3	6.5	5	5.9	8	4.9
10	4-7T, 4A			6	13.0	8	9.4	14	8.6
11	4-8T, 4A	2	6.3	1	2.2	10	11.8	13	7.9
12	4-9T, 4A	5	15.6	2	4.3	3	3.5	10	6.1
13	4-5T, 4-5A			1	2.2	2	2.4	3	3.1
14	4-6T, 4-5A			1	2.2	4	4.7	5	3.1
15	4-7T, 4-5A			1	2.2			1	0.6
16	4-8T, 4-5A	2	6.3	5	10.9	1	1.2	8	4.9
17	4-9T, 4-5A	10	31.3	2	4.3	5	5.9	17	10.4
18	4-6T, 4-6A			1	2.2	1	1.2	2	1.2
19	4-7T, 4-6A	1	3.1	1	2.2			2	1.2
20	4-8T, 4-6A			1	2.2			1	0.6
21	4-9T, 4-6A	6	18.8					6	3.7
22	4-7T, 4-7A								
23	4-8T, 4-7A			1	2.2	1	1.2	2	1.2
24	4-9T, 4-7A								
25	4-8T, 4-8A								
26	4-9T, 4-8A								
27	4-9T, 4-9A								
Classified		28	87.5	35	76.1	59	69.4	122	74.8
Unclassified		4	12.5	11	23.9	26	30.6	41	25.2
Mean Ability Z Score		1.75		-.04		-.14		0.00	

[a] T = Tonal, A = Atonal, "-" = Through

Contact: Walter Vispoel
Township H.S. District 214
799 W. Kensington Road
Mt. Prospect, IL 60056

312-259-5300

CROSS-CULTURAL REPLICATION: THE USE OF CONCRETE AND COMPUTER SIMULATION ACTIVITIES FOR LEARNING ELEMENTARY SCHOOL MATHEMATICS CONCEPTS

Dr. Arthur L. White - The Ohio State University - Columbus

Dr. Donna F. Berlin - The Ohio State University - Newark

Dr. Pamela F. Abder - The University of the West Indies - Trinidad

ABSTRACT

This study is a replication of a previous study reported in the Proceedings of the 26th International Conference of the Association for the Development of Computer-Based Instructional Systems (Berlin & White, 1985). The theoretical framework for the research is based upon modes of representation. It is suggested that learning progresses along a continuum from concrete to semiconcrete to abstract. This research was designed to investigate effects of interactive microcomputer graphics simulation activities on understanding of mathematics for West Indian elementary school children in Trinidad and Tobago. The children had little or no experience with manipulative materials or computers prior to the research. The children were in second grade (ages 6 to 7) and fourth grade (ages 8 to 9). There were 28 boys and 38 girls in the sample. The three treatment conditions involved computer only, manipulatives and computer, and manipulatives only learning activities. There were no differential effects between the use of manipulative materials and the computer. The fourth grade girls scored higher on design recognition than did boys and generally the fourth grade students did better than the second grade students. There were apparent similarities and dissimilarities in the findings of this study and that of Berlin and White (1985). In each study, the fourth grade subjects scored better on the criterion (design recognition and sequence) than the second graders. There was evidence of gender effects which were related to grade level, socio-cultural background, and/or treatment.

This study is a replication of a previous study reported in the Proceedings of the 26th International Conference of the Association for the Development of Computer-Based Instructional Systems (Berlin & White, 1985). This research was designed to investigate the effect of interactive microcomputer graphics simulation activities on the transition from concrete manipulations of objects to abstract thinking in elementary school mathematics.

The theoretical framework for the research is based upon modes of representation. It is suggested that learning progresses along a representation continuum from concrete to semiconcrete to abstract. This continuum has been translated into instructional practice as manipulation of physical objects, pictorial or graphic representation, and finally symbolic form. Activities based upon modes of representation are consistent with current learning theory and reflect instructional practices valued in the teaching of mathematics: activity-based learning, hands-on activities, learner centered, multisensory, and multimedia environments.

Recent literature continues to reveal additional documentation related to the use of instructional activities based upon the representational mode continuum. There is a robust body of literature which supports the use of concrete objects and manipulative materials in the teaching of mathematics (Penick, 1983; Reys, 1971; Suydam & Dessart, 1976; Arithmetic Teacher Focus Issue on Manipulatives, February, 1986). Reviews of the research by Suydam (1984a, 1986) and a meta-analysis by

Parham (1983) indicate that the use of manipulative materials in mathematics education enhances achievement across a variety of topics, grade levels, achievement levels, and ability levels.

The use of imagery or semiconcrete representations has also been well-documented. Paivio's (1971) dual processing hypothesis suggests an inherent ability to process information by verbal and/or visual procedures. Studies have shown that teachers who use imagery-related media facilitate achievement (Boswell, 1980; Holliday, 1975; Zimmerman & Rosenthal, 1972). Specific to the current research is the literature on the instructional use of computer-generated images. In reviews of the research, Burns (1982) and Suydam (1984b) found that both achievement and positive attitudinal effects were associated with the use of computers in mathematics education. Computer-based drill and practice and tutorial programs tended to be less time consuming and more effective than traditional mathematics instruction. There have been a number of recent meta-analytic reviews which document the benefits of computer-based education (CBE). Bangert-Drowns, Kulik, and Kulik (1984) and Kulik (1983) found that CBE has the potential to improve student achievement; promote a positive attitude toward the subject matter, instruction, and computer technology; and decrease instructional time. Reviews of CBE literature by both Bracey (1982) and Ploeger (1984) report positive achievement, motivation, and self-esteem effects associated with computer-based instructional activities. Ploeger suggests that the computer is most effective as an adjunct to traditional instructional methods. The current research focuses on the effect of manipulation of concrete objects and/or computer simulations on more traditional paper-and-pencil activities.

Statement of the Problem

The problem in this study was to investigate the effect of interactive microcomputer graphics simulation activities on the transition from concrete manipulations of objects to abstract thinking in elementary school mathematics.

Objectives

Specifically, the study will assess student performance:

1. In recognition and selection of duplicate designs.
2. In recognition of sequences, rule identification, and extension of patterns.

METHOD

Subjects

The study involved 66 second and fourth grade children from an Indian School in San Fernando, Trinidad and Tobago in the West Indies. The school is made up of children who have grown up in the Indian culture and whose parents or grandparents are descendants from East India. The children had little or no experience with manipulative materials or with computers prior to the research. The children represented two age levels (6 - 7 years) and (8 - 9 years). There were 28 boys and 38 girls in the sample. The children were randomly assigned to three treatment conditions.

Treatment

The treatment conditions varied on the representational level (concrete or semiconcrete) of the learning activities. The computer group completed ten activities using a computer simulation of color cubes to reproduce designs and sequences which were presented to them by use of task cards. The manipulative/computer group completed five activities using manipulative materials (colored cubes) followed by five activities using the computer simulation to reproduce designs and sequences which were presented by use of task cards. The manipulative group completed ten activities using the colored cubes to reproduce designs and sequences which were presented by use of task cards. Each subject worked individually and completed two to three task cards at one time. The children were scheduled for the activities over a two to three week period.

At the end of the treatment period, a paper-and-pencil instrument consisting of two, six-item scales was administered and subjected to item analysis. Scale 1 (Sc1) is a measure of the childrens' ability to recognize a design and pick it out from among a set [illegible] distractors (mean difficulty = .77, mean discrimination = .63, KR-20 reliability = .62). Scale 2 (Sc2) is a measure of the childrens' ability to recognize a sequence, identify the rule for the sequence, and predict the next element in the sequence (mean difficulty = .33, mean discrimination = .60, KR-20 reliability = .63).

The means and standard deviations for these scales are given in Table 1.

Table 1. Means and Standard Deviations by Cell for Scale 1 and Scale 2

Grade		Gender							
		Boys				Girls			
		2nd		4th		2nd		4th	
		Sc1	Sc2	Sc1	Sc2	Sc1	Sc2	Sc1	Sc2
Treatment Computer	M	1.0	3.0	1.1	4.0	0.0	3.6	3.4	5.1
	SD	1.0	1.0	1.1	1.6	0.0	1.5	1.2	1.4
	(N)	(3)		(7)		(5)		(8)	
Manipulatives & Computer	M	0.0	2.8	1.6	3.4	0.0	4.0	1.2	4.8
	SD	0.0	0.5	1.7	1.1	0.0	1.7	1.0	0.8
	(N)	(4)		(5)		(3)		(9)	
Manipulatives	M	1.5	2.5	1.4	3.9	0.3	3.3	2.6	5.1
	SD	0.7	0.7	1.1	1.5	0.8	1.6	1.9	0.7
	(N)	(2)		(7)		(6)		(7)	

These two scales were used together as separate criteria in a 3x2x2 three way multivariate analysis of variance to test for significant interaction effects between the Treatments, Gender, and Grade Level as well as for main effects. The analysis revealed the following:

1. **No Treatment by Gender by Grade Level effect was found.**
2. **A significant (p<.02) Gender by Grade Level interaction effect was found. On Scale 1 the scores for the girls increased from the 2nd grade to the 4th grade more than did the boys scores.**
3. **No Treatment by Gender effect was found.**
4. **A significant (p<.001) Grade Level effect was found. The 4th grade children did better on both scales than did the 2nd grade children.**
5. **A significant (p<.008) Gender effect was found. The girls did better overall than did the boys, especially on Scale 2.**
6. No significant Treatment effect was found.

The multivariate analysis of variance is summarized in Table 2.

Table 2. MANOVA: Multivariate Tests

Effect	p<
Treatment x Gender x Grade Level	.21
Gender x Grade Level	.02 *
Treatment By Grade Level	.70
Treatment by Gender	.35
Grade Level	.001 ***
Gender	.008 **
Treatment	.29

* p < .05
** p ≤ .01
*** p ≤ .001

The univariate analyses of variance for Scale 1 and Scale 2 are given in Table 3. These analyses are only reported for effects where significant (p<.05) multivariates effects were found. The alpha level probabilities are totally consistent with discriminant function coefficients for the multivariate analysis.

Table 3. ANOVA: Univariate Tests

Effect		F	MS	p<	DF
Gender by Grade Level					
	Scale 1	8.14	10.67	.006	[illegible] **
	Scale 2	.29	.47	.59	- [illegible]
Grade Level					
	Scale 1	29.03	38.04	.001	.74 ***
	Scale 2	14.69	23.70	.001	.47 ***
Gender					
	Scale 1	1.80	2.36	.19	.21
	Scale 2	10.23	16.51	.002	.93 **

* p ≤ .05
** p ≤ .01
*** p ≤ .001

The means for the Gender by Grade Level interaction and the main effects for Gender and Grade Level for Scale 1 and Scale 2 are given in Table 4. A plot of the means for the Gender by Grade Level interaction for Scale 1 (p<.006) is given in Figure 1.

Table 4. Gender, Grade Level, and Gender by Grade Level Means

Gender	Grade Level					
	2nd		4th			
	Sc1	Sc2	Sc1	Sc2	Sc1	Sc2
Male	.7	2.8	1.4	3.8	1.1	3.5
Female	.1	3.6	2.3	5.0	1.5	4.5
	.4	3.3	1.9	4.5		

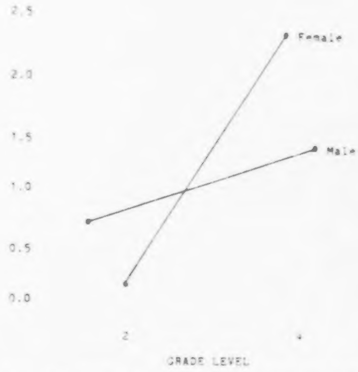


Figure 1. Gender by Grade Level Interaction for Scale 1

Discussion

It is clear from the results that 4th grade students are better able to orient and recognize designs and determine and apply rules for sequences than are 2nd grade children. It is also apparent that these skills are dependent upon Gender in this West Indian population. The girls appear to develop these capabilities at an earlier age than do the boys and by the 4th grade the girls are significantly better at these tasks than are the boys. Further research for the purpose of better defining the nature of the influences of technology on differential learning efficiences for boys and girls in this population is appropriate.

The data from this study with subjects in an all West Indian-urban population in Trinidad and Tobago and the data from Berlin and White (1985) from an all white-rural population and an all black-suburban population were the result of the same treatment procedures and activities. There were apparent similarities and dissimilarities in the findings.

In each setting the fourth grade subjects scored better on the criterion than the second graders. The criterion apparently measures skills and/or understandings which develop with time either through maturation and/or learning. In each setting there was evidence of gender effects which were generally confounded by significant interactions with grade level, socio-cultural background, and/or treatment. In the white-rural setting, the girls tended to do better with concrete materials while the boys did better whenever the computer was involved. In the black-suburban setting, the girls using the concrete materials did not do well at all. In the West Indian-urban setting the girls, particularly at the fourth grade level, did much better than the boys regardless of whether they used concrete materials, the computer, or a combination of concrete materials and computer activities.

The questions concerning the interactive nature of these tasks with gender, grade level, and socio-cultural background are many and not easily answered. The findings more germane to the development, refinement, and validation of instructional models are those involving the treatment interaction effects. It appears, at this point, that the treatments have little or no effect on that the power of the treatments has not been sufficient to override the other factors such as gender, socio-cultural background, and grade level. For future research the following directions will be pursued.

A. The treatments will be modified so as to cover a greater time span and give the subjects a more intensive experience with the activities.
B. The nature of the combined concrete materials and computer activities will be changed to provide more linkage between the two modes.
C. The criterion instrument will be modified to more closely relate to the skills and understanding for which the activities were designed.
D. Subjects will be drawn from a white-rural population similar to the White and Berlin (1985) study where significant treatment interaction effects were found.
E. A sampling of the subjects used in the research will be interviewed so as to gain insights and perspectives from the learners concerning the impact of the concrete materials and the computer on their learning processes.

References

Arithmetic Teacher Focus Issue Manipulatives. (1986, February). Reston, Virginia: National Council of Teachers of Mathematics.

Bangert-Drowns, R.L., Kulik, J.A., & Kulik, C.C. (1984, May). The coming of age of educational technology: Meta-analyses of outcome studies of computer-based instruction. Proceedings of the 25th International Conference of the Association for the Development of Computer-Based Instructional Systems, Columbus, Ohio.

Berlin, D.F., & White, A.L. (1985, March). The computer as transition from object manipulation to abstract visuo-spatial thought. Proceedings from the 26th International Conference for the Association for the Development of Computer-Based Instructional Systems. Philidelphia, Pennsylvania.

Boswell, D.A. (1980). Evaluation of transparencies for psychology instruction. Teaching of Psychology, 7, 171-173.

Bracey, G.W. (1982, November/December). Computers in education. What the research shows. Electronic Learning, 51-54.

Burns, T.K. (1982). A quantitative synthesis of research findings relative to the pedagogical effectiveness of computer-assisted mathematics instruction in elementary and secondary schools. Dissertation Abstracts International, 42A, 2946.

Holliday, W.G. (1975). The effects of verbal and adjunct pictorial-verbal information in science instruction. Journal of Research in Science Teaching, 12, 77-83.

Kulik, J.A. (1983, October). Effects of computer-based teaching on learners. Paper presented at the meeting of the National Forum of the College Board Symposium on Computer Competency and the Curriculum, Dallas, Texas. (ED 246877)

Paivio, A. (1971). Imagery and verbal processes. New York: Holt, Rinehart, & Winston.

Parham, J.L. (1983). A meta-analysis of the use of manipulative materials and student achievement in elementary school mathematics. Dissertation Abstracts International, 44A, 96.

Penick, J.E. (Ed.), (1983). Focus on Excellence. Vol. 2, No. 1. Elementary Science. Washington, D.C.: National Science Teachers Association.

Ploeger, F.D. (1984, December). The effectiveness of microcomputers in education. Southwest Educational Development Library. (ED 246 8761).

Reys, R.E. (1971). Consideration for teachers using manipulative materials. Arithmetic Teacher, 18, 551-558.

Suydam, M.N. (1984a). Research report: Manipulative Materials. Arithmetic Teacher, 31, 27.

Suydam, M.N. (1984b). Research report: Microcomputers in mathematics education. Arithmetic Teacher, 32, 35.

Suydam, M.N. (1986). Research report: Manipulative materials and achievement. Arithmetic Teacher, 33, 10 & 32.

Suydam, M.N., & Dessart, D. (1976). Classroom ideas from research on computational skills. Reston, Virginia: National Council of Teachers of Mathematics.

Zimmerman, B.J., & Rosenthal, T.L. (1972). Concept attainment, transfer, and retention through observation and rule-provision. Journal of Experimental Child Psychology, 14, 139-150

Contact: Arthur L. White
The Ohio State University
252 Arps Hall
1945 North High Street
Columbus, OH 43210

COLLABORATION IN THE DESIGN AND TEACHING OF COMPUTER CONTINUING NURSING EDUCATION

Carol A. Bauer*

Morristown Memorial Hospital

The knowledge that nurses need about computers is affected by their daily roles and responsibilities. These vary from physiological monitoring for the critical care nurse to budget management by the nursing administrator. Collaboration between nursing service and nursing education occurred through meetings of two colleagues representing nursing education and nursing service. These collaborative efforts began with exchange of activities and then realization that resources of hospital information system at acute care agency and microcomputer laboratory at educational institution were best utilized through joint continuing education efforts. The basic computer course includes ten class sessions which focus on general literacy in word processing, data processing, graphics and spread sheets and related nursing applications and issues. Content for the advanced course was based on survey results conducted among 445 nurse educators/managers in northeast. These findings indicated that there are commonalities and significant differences in needs for computer learning experiences among these two groups. Classroom and laboratory content for basic and advanced computer classes will be described. Collaboration between nursing education and nursing service has and will continue to influence the integration of computer knowledge within nursing practice and education.

*Morristown Memorial Hospital
100 Madison Avenue
Morristown, NJ 07960
201-540-5452

COMPUTER ASSISTANCE IN
FOREIGN LANGUAGE INSTRUCTION

Howard Jeffrey Bender*

MindBender Associates

This presentation will be on the first phase of a continuing research project to translate English to other languages. Results will be reported on the Feasibility Study, the Functional Requirements Study, and the System Specification aspects of the project.

The Feasibility Study addresses the linguistic difficulties found in other languages. Lexical ambiguities such as "I like squash" (vegetable or game?), knowledge ambiguities such as "the hamburger is ready to eat" (hamburgers do not eat), and structural ambiguities such as "he left his money in the car, but it was stolen" will be discussed. Approaches to output rendering of complex scripts, such as Arabic (right to left), Japanese (mixture of ideographs and phonetic letters), and Mongolian (vertical) will be included. Data base sizing will be discussed.

The Functional Requirements Study will report the language translation needs of users in travel, education, diplomacy, and international trade. Possible personal computer configurations will be compared, with deficiencies and limitations noted.

If time permits, a proposed top-level software system design will be discussed. Data flow diagrams and system flow charts will be included.

This research is being funded by a U.S. Department of Education Small Business Innovative Research grant.

*MindBender Associates
7716 Hanover Parkway, No. 104
Greenbelt, MD 20770

301-474-1938

THE TRI-LEVEL APPROACH TO CBT

Norm Carr, Geoff Burston - Ontario Hydro

Michael Szabo* - The University of Alberta

Ontario Hydro is a power utility with 25,000 employees. This large corporation must continually ensure that its operating and support staff competently perform their jobs and requires a large commitment to performance based training for skills and knowledge maintenance.

An automated method of competence assessment, feedback, prescription generation and training delivery is essential to meet our demands and maintain an acceptably high level of performance.

An evaluation of existing CBT systems indicated that no single decentralized system fully meets the diverse needs of big training populations. Flexibility requires the use of multiple systems and languages, co-existing in the same environment; the most appropriate used for each specific application. This new initiative was called the TRI-LEVEL approach since it integrates the mainframe, local area network and micro environments.

TRI-LEVEL adopts industry standard (MS-DOS, UNIX) hardware, and variants of a highly developed authoring language which operates in all three environments. TRI-LEVEL is driven largely by changes in the nature of work and the mobility of the work force. The TRI-LEVEL approach has been formulated to provide upward compatibility with new trends, such as the establishment of free form learning databases (visual, text, audio, etc.) versus the "traditional structured curriculum" approach to training.

Although implementation of this TRI-LEVEL system is in progress, evaluation indicates this approach holds the most promise for future CBT initiatives.

*700 University Avenue
Toronto, Ontario
Canada

416-592-5111

MAKING INTELLIGENT CHOICES AMONG MICROCOMPUTER-BASED AUTHORING SYSTEMS

Victoria L. Clegg, Jane M. Lommel*

Kansas State University

Microcomputer-based authoring systems are one of the major tools available to create CBI courseware. These systems incorporate instructional design capabilities and, in contrast to authoring languages and programming languages, require a minimum knowledge of programming. What corporate trainers and educators have needed is a comprehensive set of evaluation criteria to help choose an authoring system that is best for them.

The Project to Evaluate Authoring Systems at Kansas State University has established a set of such criteria by gleaning expert opinion through the Delphi technique. Opinions about the major characteristics of authoring systems were gathered from two groups: (a) experts in computer-based instruction and (b) senior product managers of firms developing authoring systems. These same participants reviewed a compiled list of the characteristics and rated their relative importance. A third group, knowledgeable users of authoring systems, also rated the list of characteristics. In a third round, all three groups were given the opportunity to refine the results of the compiled ratings.

To help in the process of establishing these criteria, the following authoring systems were either donated or loaned to the Project as of April 1, 1986: ADROIT from Applied Data Research, PLATO from Control Data Corporation, CBT from ForceTen Enterprises, Scholar-Teach from Boeing Computer Services, TenCORE from Computer Teaching Corporation, InterAct from Ashton ITC, and TAIGA from Twente University of Technology.

The ADCIS session will concentrate on descriptions of the criteria, their relative importance as indicated by the three groups of participants, and the Project methodology. The criteria were developed so that people can use them and will be distributed to session participants.

*Kansas State University
Fairchild Hall, Room 215
Manhattan, KS 66506

913-532-5712

PRODUCTION SYSTEM BASED COMPUTER-ASSISTED INSTRUCTION

S. C. Davis, D. Sullivan,* M. West

Social Security Administration

The Social Security Administration (SSA) currently uses a large, centralized data processing system in the daily conduct of its business. This system provides technicians and managers in 1,400 offices across the country with online access to over 350 million individual benefit payment and earnings records. Millions of interactive transactions are processed against these records each day.

The necessity to fully utilize all available resources in meeting numerous training needs resulted in the establishment of this computer-assisted instruction (CAI) development project. Its principal objective was to use SSA's large mainframe production system to design, develop, and deliver CAI formatted materials to our professionally diverse and geographically dispersed 70,000 member audience. These CAI materials are needed to support numerous training/retraining needs associated with SSA's ongoing systems modernization efforts. This continuous modernization is necessary for SSA to maintain its outstanding level of service to the public.

A project team composed of instructional technologists, automated data processing (ADP) professionals, and a private sector consultant with CAI/CBT experience was assembled. The project began with an analysis of the situation followed by a feasibility recommendation for using our mainframe production system in delivering CAI based materials. The inclusion of an impartial third party (consultant) viewpoint strengthened our recommendation that a production system based CAI program was the best delivery approach.

*Social Security Administration
4400 Annex
6401 Security Boulevard
Baltimore, MD 21235

301-594-7486

MACHINES THAT TEACH:

A HISTORY OF COMPUTER-BASED INSTRUCTION

Brian L. Dear*

Hazeltine Corporation

The history of computer-based instruction can be traced back to the early part of the twentieth century when Sidney Pressey and later B. F. Skinner worked on mechanical teaching machines. By the early 1960s computers replaced the mechanical devices and the PLATO and IBM Coursewriter projects had begun. Throughout the 1960s the technology improved and CBI research expanded to many sites nationwide. The TICCIT system was born in the late 1960s and was developed in the early 1970s; WICAT followed later in the decade.

Each of these systems evolved differently; each has its own story. This presentation reports on my current research for a book about the history of the CBI field: a book that tries to capture the mix of people, ideas, forces at work and spirit of the times that led to the CBI systems of today.

It would be impossible to cover the entire history of the field in one brief conference session. Therefore, this session offers a selection of PLATO, TICCIT, IBM, WICAT and other stories from the manuscript - a wide selection of anecdotes covering a wide and varied history.

*Hazeltine Corporation
10800 Parkridge Boulevard
Reston, VA 22091

703-602-6800

SCREEN DESIGN: CREATIVE USES OF SIMPLE LINE MONOCHROME GRAPHICS

J. D. Ellertson*

Nationwide Insurance

The media/convention blitz surrounding the need for "advanced color graphics" can sometimes be overwhelming for new course writers, especially if the company has already selected the hardware/authoring system and simple line monochrome graphics are the reality of the day.

Rather than being concerned with "wouldn't it be nice" issues, course writers need to generate quality instructional programs within their system's constraints. To avoid that dreaded "electronic page-turning" syndrome, course writers must present concepts in terms of thought patterns, build blocks of information into sequences of ideas and create visual layouts that both invite and hold the student's attention.

The CAI Division at Nationwide Insurance uses the Burroughs B-25 and the EDUCATOR authoring system (Spectrum Taining Corporation, Version 2.0, simple line monochrome graphics). In the process of developing instructional programs for agent training, creative ways have been developed to break up "white space", write copy to fit single line boxes, and block ideas through a series of windows to form a total screen design impact. Pilot tests quickly indicated that simple line monochrome graphics can have a dramatic effect in creating visually attractive and effective computer based learning.

*Nationwide Insurance
CAI Division
One Nationwide Plaza - 29T
Columbus, OH 43216

614-249-6229

A PROGRAM TO REDUCE ON-THE-JOB TRAINING FOR AIR TRAFFIC CONTROLLERS THROUGH COMPUTER-BASED INSTRUCTION

Doug Farrow*

Logicon, Inc.

Air Traffic Controllers graduating from the Federal Aviation Academy currently require an additional 250 hours of radar training on-site at each Air Route Traffic Control Center (ARTCC) before being certified to control live air traffic unsupervised. This training consists of 10-15% computer-based simulation and 85-90% supervised on-the-job training (OJT). This OJT is conducted on the control room floor using live traffic and a one-to-one student-to-instructor ratio. This is expensive, time-consuming and stressful for students, instructors and pilots alike.

In an effort to improve this difficult training environment, the Seattle ARTCC has developed a number of software upgrades to the existing simulation capabilities. The goal has been to shift as much training as possible from OJT to simulation. During a recent pilot test, this new Computer Enhanced Radar Training (CERT) program reduced total training time by over 25%. The increased use of computer support also reduced the instructor requirement, increased safety, decreased stress and reduced the calendar time required for student certification by approximately 50%.

*Logicon, Inc.
475 School Street S.W.
Washington, D.C. 20024

202-646-5466

DOE CONTRACTOR RADIATION SAFETY CBT COURSE

Patrick R. Gardner*

Westinghouse Hanford Company

This is a completed project report. Westinghouse Hanford Company developed a generic Radiation Worker safety CBT course for actual and potential tasks and included visits to fourteen different contractor sites. Team Design and Prototype verification formed the major portion of the development phase. Lesson entry was accomplished using the WISE author system from WICAT Systems, Inc.

Released in April this year, the course features: (1) graded task simulations for both Pretest and Final, (2) fourteen Topics in five lessons, each Topic keyed to "Critical Acts" and Questions in the Pretest and Final, (3) Automatic, Intensive and Manual modes of instruction available for each lesson, (4) Practical Problems and Sample Questions associated with each Topic, (5) provisions for local configuration in several areas. The course is deliverable on IBM PC compatible equipment and will be available for review at the Conference.

*Westinghouse Hanford Company
P.O. Box 1970
Richland, WA 99352

509-376-0044

CBT IMPLEMENTATION

A RISKY BUSINESS

Lori Gillespie *

Instructional Communications, Inc.

There are many risks inherent in making a change...any change...in the way an organization operates. Implementing CBT involves multiple changes to all existing organizational systems, compounding the risk of failure.

The first changes made when implementing CBT are to the training system itself. Adding CBT to an existing training system will affect how courses are designed and developed, as well as how they are produced, distributed and delivered. These changes in turn affect the training and overall organizations' performance systems by altering how organizational divisions interact with each other and the roles of the personnel in those divisions. This usually requires changes to staffing and training requirements and organizational administrative policies and procedures. Physical changes include the addition or renovation of space, furniture and equipment to meet the new delivery, production and administrative requirements. The economic system requirements change as more money is needed for course development and less for course delivery.

This presentation will describe the potential obstacles to be faced when implementing CBT and all its changes; the risks brought by these obstacles and the failure to deal with them; and suggested activities to reduce risk of failure.

*1902 Eutaw Place
Baltimore, MD 21217

301-728-8763

INSTRUCTIONAL TECHNOLOGY WITHIN THE SUPERCOMPUTING ENVIRONMENT:

USING CBT AS THE INSTRUCTIONAL MEDIUM

Stephanie S. Goodwin*

NASA Ames Research Center

In the last several years, Computer-Based Training (CBT) has proven itself as a viable solution to the instructional problem of providing instruction within the computing environment. Whether or not the full potential of this tool has been realized remains to be seen. With the advent of newer technology comes new applications. The development of a newer technology (supercomputers) provides for a new application for the use of CBT, that will truly test the value and effectiveness of this medium as an instructional tool. The purpose of this presentation is to outline and discuss the rationale and implications for using CBT as the instructional medium within a newer technological environment; the supercomputing environment.

This presentation will primarily deal with describing the differences between past and existing computing environments with the newer supercomputing environment, in terms of instructional problems. We will briefly describe the technology and environment of supercomputing, with an emphasis on the rationale that makes the use of CBT so appropriate for this environment. This rationale will include hardware and software considerations, but more importantly, an in-depth description of the user community (location, sophistication, needs and requirements, etc.) as it relates to the media selection activity within the Instructional System Development (ISD) process.

*NASA Ames Research Center
Mailstop 223-2
Moffett Field, CA 94035

415-694-6477

HAVE YOU HAD YOUR TRAINING PROGRAM EVALUATED LATELY?

B. H. Handler*

Oak Ridge National Laboratory

Most evaluation of CBT programs is done in the form of performance measurement, if it is done at all. The users of a program are an untapped source of valuable information concerning their training. Before a program is designed, the authors define the audience and its traits, but after the program is designed, little attention is paid to whether the user needs were met. This presentation will explain an on-line evaluation procedure developed for a training program for the U.S. Navy and easily applied to CBT in general.

After the user completes the last phase of on-line training, a series of evaluation questions are presented with a linear rating scale. The user is to respond to the question with a numerical rating; however, there is a place for free form answers regarding that particular question. After all of the author designed questions have been displayed, the user is allowed to make any other comments he/she feels are appropriate. All of the responses are written to a file and may be examined by the supervisor, manager, trainer, and/or author whenever it is convenient for them to do so. The user responses should be considered in updating the training package.

---

*Oak Ridge National Laboratory
Oak Ridge, TN 37831

615-574-0262

MANAGING THE COURSEWARE DEVELOPMENT PROCESS

Ruth M. Hawkins*

McDonnell Douglas Corporation

Even though the courseware development process is an expensive and time consuming one, an astute project manager can minimize this expense without sacrificing quality. Careful planning, staffing, scheduling and controlling of the project will help to avoid development pitfalls that cause schedules to slip and budgets to overrun.

This presentation will address the following areas that concern courseware development project managers:

Project scheduling
- The project schedule is influenced by such factors as customer review cycle, available development staff, complexity of content and customer/user requirements.

Staffing
- To staff a new development project, a manager must hire and train new staff or allocate existing staff.

Establishing development standards and quality assurance
- In order to obtain a high standard of quality and consistency, standards need to be established for the development process including a quality-assurance check to verify that standards have been met.

Establishing the customer/user interface
- Many delays in development can be avoided by clearly establishing the nature and amount of customer/user involvement in the development process. For example, design reviews and product approval procedures should be established at the outset of a project.

Managers can avoid many of the schedule and budget problems that befall many courseware development projects if they anticipate these issues and plan effectively.

*1390 S. Potomac, Suite 124C
Aurora, CO 80012
303-337-6303

TEACHING A PROBLEM SOLVING STRATEGY
VIA COMPUTER INTRODUCING "EEV"

Bonney Hettinger*

University of Maryland/University College

The computer's analytical capabilities make it an ideal tool for solving problems which are complex and time-consuming OR for teaching the learner to use a new problem solving strategy.

During the design of a CBI course in Nuclear Technology, it became evident that the method used to solve typical problems for this introductory course was more important than the answer. "EEV", an expression evaluator, was designed to teach the learner the new problem solving strategy.

The Expression Evaluator (EEV) is designed to be used in situations where the answer that the student arrives at for a question is not as important (or at worst equally important) as the method the student used to get the answer.

The system allows the student to enter the answer (with the correct units) at any time during the problem, or, if the student does not know how to proceed, he can choose to substitute an equation for part of the answer. The student does not need to enter the answer in its simplest form, as the system simplifies an answer the student enters for him. The system tracks the derivation process that the student uses to answer the question, and the designer can develop very effective instructional feedbacks.

*University of Maryland/University College
University Boulevard at Adelphi Road
College Park, MD 20742

301-985-7356

COMPUTER-BASED TRAINING FOR
CHILD PROTECTIVE SERVICE

A. N. Hewitt*, C. LaBrosse, N. A. O'Neill

Western Washington University

Child Protective Service Workers have been severely criticized nationally for being inexperienced and inconsistently trained in the investigation of child abuse and neglect. Initial training is often delayed due to variable training schedules and high caseloads, resulting in minimally trained personnel making crucial decisions affecting the lives of children and families.

The Center for Higher Education at Western Washington University, through the Program for Social Service Research, Demonstration and Training, has developed a six-hour, self-study computer-based training for child abuse investigators. The program, "Child Abuse and Neglect" is designed to teach patterns consistent with child maltreatment.

The training is delivered at the work station by a level III videodisk program during the first week of employment. This information is generally provided by supervisory consultation, program manuals and conventional training.

This project, funded by the Department of Health and Human Services, introduces computer-based training to an audience not generally exposed to computers in the workplace. The program was developed using the authoring language PILOT II. The presenters were responsible for selection and purchase of hardware and software, content development, programming and video production.

The presenters will demonstrate the program, outline the project development, and describe field testing and evaluation results.

*Seattle Training Center
1602 N.E. 150th, N17-3
Seattle, WA 98118
206-464-6799

DESIGN OF A COMPUTER APPLICATIONS SURVEY COURSE

Norman Higgins*

Arizona State University

This presentation describes the design of a graduate level computer applications survey course. The course is designed to expose students through hands-on-experiences to a variety of computer applications: word processing, graphic design, instruction, data bases, authoring languages, etc. Students are expected to be able to run at least two programs in three application areas and four programs in a fourth application area. Criteria for the selection of application areas and computer programs for each area are described in this presentation.

The course is personalized to the extent that students are able to select the application areas that interest them. They are also able to select the programs that they will run and evaluate for specific application situations they specify. An adaptive test designed to assess the students ability to transfer their knowledge of specific computer programs to various application situations is described. Other course assignments designed to engage students in the identification, selection and evaluation of computer programs are described.

Sample course materials for one application area, e.g., Graphics Design, are used to illustrate the design features of the course.

*College of Education
Arizona State University
Tempe, AZ 85287

602-965-3651

COMPUTER SIMULATED LOGIC CIRCUIT INSTRUCTION

Jon-Chao Hong*

National Taiwan Normal University

The "Logic Circuit" software was developed for students who want to become familiar with and understand the functions of logic gates and logic circuits.

Therefore, every type of logic gate and those with TTL IC symbols were designed in this software. Users can envisage their own idea and develop a logic circuit by selecting and connecting gates which execute the real situations and conditions immediately. It also allows the users to manipulate the clock and to observe the out-put signals and truth table of logic circuits.

The most important traits of this software are to save manipulating time in circuit connecting and executing and to avoid the modeling materials which are used in traditional instruction.

*National Taiwan Normal University
Department of Industrial Education
Taipei, Taiwan
(02) 3219742

EMBEDDED TRAINING:

THE EXPLOSIVE GROWTH AREA IN CBT

Julie E. Horine,* Jeffrey B. Erickson

Arthur Andersen and Company

An important factor for successful implementation of any computer system is the efficient, effective training of system users. User training typically involves "hands-on" use of actual system software, an approach which increases the transfer from training to job performance. Another approach to user training is simulation of system software using computer-based training (CBT). The CBT approach provides consistent training to all users with increased control over the training process. And, with large numbers of users and long course life, CBT can be a cost-effective approach to user training.

Embedded training, the concept of physically or logically integrating computer-based training with actual system software, combines the effectiveness of "hands-on" software training with the increased consistency, control and cost-effectiveness of computer-based training.

Arthur Andersen and Company recently completed a project to develop embedded training for 15,000 users of a new nationwide computer system. This project included a review of embedded training design standards, prototype design and development, and user training design and development.

This presentation will focus on: 1) defining two major approaches to embedded training, 2) discussing types of training that may be embedded, 3) demonstrating commercial and custom-developed embedded training products, and 4) discussing issues in the design, development, delivery, and maintenance of embedded training. The difference between designing traditional CBT and designing embedded training will also be examined. Finally, a discussion of key success factors for embedded training will be presented.

*Arthur Andersen and Company
1666 K Street NW
Washington, D.C. 20006
202-862-3386

DISCREPANCY EVALUATION MODEL FOR COMPUTER-BASED EDUCATION

David L. Jelden*

University of Northern Colorado

The Learner-Control Education System (LCES) is best described as an individualized computer-managed instructional sequence of activities which utilizes a wide variety of media. LCES basically consists of four interacting modules: a) Student Information Module (SIM) contains the essential data relevant to understanding a student's academic strengths, weaknesses, experience, expectations and goals; b) Instructional Analysis Module (IAM) generates the specific content of the program and orders it for instructional purposes according to a psychological/logical order using criterion objectives; c) Learning Activity Module (LAM) contains those activities which constitute the actual learning sequences for each student (using all available support media and CAI) as well as the computer management of learning options; and d) System Evaluation Module (SEM) provides the taxonomy of assessment techniques and methodologies used to measure the degree to which LCES optimizes learning in an economical and efficient manner.

The purpose of this paper is to describe in summary the system evaluation module (SEM) of Learner Controlled Education. More explicitly, the various assessment problems most frequently encountered with such comprehensive computer based education systems.

---

*University of Northern Colorado
Educational Technology Center
Greeley, CO 80639
303-351-2186

DIRECT MANIPULATION OF LESSON FLOW STRUCTURE

Robert L. Jorczak*

Control Data Corporation

Traditional courseware development tools provide little or no aid in establishing the general structure (or strategy) of a CAI lesson. A new authoring system, PCD3, has a specialized utility that provides a pictorial representation of lesson strategy and allows the author to directly manipulate the strategy.

For the first time authors, instructors and researchers will be able to see and easily change CAI lesson structure. This feature can decrease lesson development and maintenance cost and time. It can lead to better lessons as lesson design flaws will be more apparent. It can aid research into CAI lesson design as lesson strategies can be classified and evaluated based on visible lesson strategy characteristics.

Independent development of lesson structure also leads to a clean separation of lesson content from structure. This separation leads to further gains in productivity as content may be developed for use by several different lesson structures (or vice versa). Such a separation of lesson content and structure is supported by other features of the PCD3 authoring system.

This presentation will discuss the theory and operation of separate lesson strategy and content in general, and will specifically show how PCD3 handles these lesson components.

*4149 11 Avenue South
Minneapolis, MN 55407
612-375-8479

SEPARATING COURSE LOGIC AND COURSE CONTENT DEVELOPMENT
IN AN AUTHORING SYSTEM

Steven Just *

AT&T Bell Laboratories

Typically, authoring systems provide many tools to assist the author in the development of course content, usually through "templates" that permit development of standard frame types (e.g. multiple choice or short answer questions.) Often, however, authoriing systems provide few tools to assist the author in the development of a course structure. Most authoring systems assume that unless directed otherwise, frames are implicitly sequenced in the order that they are created. Thus, many authors fail to develop an explicit instructional strategy for their courses, concentrating almost exclusively on frame content. AT&T's UNIX System V INSTRUCTIONAL WORKBENCH Authoring System overcomes this weakness by requiring the author to specify an explicit course structure and by providing course logic "templates" to assist the author in accomplishing this task.

In IWB, course content and course logic are created entirely independently, but both are required. Authors usually begin by specifying the course structure first and then detailing frame content; however, there is no implied order to the development process and most authors find themselves working iteratively, initially developing a broad course structure, then filling in frame content, then refining the structure, etc.

This talk will discuss the advantages of an explicit logic/content separation and its implementation in the IWB authoring system.

*10 Empress Lane
Lawrenceville, NJ 08648

201-981-2045

EXPLORATIONS IN THE EVALUATION OF AUTHORING SYSTEM SOFTWARE

Craig Locatis,* Victor Carr

National Library of Medicine/
National Institutes of Health

The purpose of this presentation is to discuss problems and procedures in evaluating authoring system software from a consumer perspective. We will review our efforts to describe system capabilities and devise procedures to assess productivity gains that systems might produce. We address assumptions underlying system development and use as well as future directions in authoring systems and their evaluation.

Rather than make gross comparisons and ratings among systems, our approach has been to develop tools that assist in local decision making and that help would-be users address the following three assumptions: that the systems will be used, that courseware quality will be improved, and that authoring will be more efficient. Each of these assumptions has varying degrees of empirical support.

We have developed an evaluation protocol which we refined in the course of reviewing over two dozen authoring systems. The protocol is predicated on first assessing authoring needs. It addresses key capabilities and features of most contemporary authoring systems that should be assessed for relevance to these needs. It also focuses on related factors affecting selection such as costs and contract terms. The protocol is useful in capturing information when viewing expert demonstrations to identify systems that may be candidates for tryout.

We have experimented with alternatives to benchmark lessons as means for assessing system performance during tryouts. Benchmark lessons have a number of limitations that our methods partially overcome by combining the identification of lesson elements with the application of a keystroke level model, devised at Xerox PARC, for estimating the efficiency of interactive computer systems. However, these methods have their own limitations.

Systems are needed that produce more intelligent courseware and that provide more authoring advice and assistance. Methods for evaluating this "next generation" of software are needed. We have compiled information on eighty systems as a result of our reviews. However, since the technology advances so rapidly written resources are outdated at the time of publication. Consequently, a data base of system information is needed that is easily up dated and widely accessible.

*Lister Hill Center
Bethesda, MD 20894

301-496-6280

COURSEWARE DEVELOPMENT FOR THE INSURANCE INDUSTRY

John A. Merrill*

Nationwide Insurance

Nationwide Insurance (headquartered in Columbus, Ohio) currently uses its own Agency Office Automation (AOA) System to enable its agents and sales managers to conduct daily business from virtually anywhere in the country via a store-and-forward computer network linked to the Home Office. In Spring 1985, Nationwide's Office of Marketing Services created the Division of Computer Assisted Instruction to develop courseware for use on the AOA System. The goals of the division are 1) to produce high-quality courseware, 2) to decentralize the training function, and 3) to reduce conventional training costs.

The creating of the CAI Division is unique within the insurance industry because it is dedicated exclusively to the training needs of the agency sales force. The courseware, which currently encompasses auto, homeowners, and life insurance, is used to train novice multi-line agents and to upgrade experienced agents on new products and services. This strategy is designed to increase product knowledge and sales activity while reducing the time spent in conventional training.

A team approach to the courseware production integrates instructional design and courseware authoring expertise with in-house subject matter experts. All courses are thoroughly evaluated and field tested prior to implementation in the Regional Offices. The EDUCATOR by Spectrum Training Corporation was chosen as the courseware authoring tool for its flexibility and potential for broad applications.

*Nationwide Insurance
One Nationwide Plaza - 29T
Columbus, OH 43216

614-249-6242

STREAMLINING GRAPHIC PRODUCTION
THROUGH REUSABILITY

Mary G. Miller*

Virginia Tech

Today many micro-based interactive videodisc systems provide users with the capability of overlay motion and stills with graphics. With this increased capability, the number of graphics used within a presentation can increase substantially. Because graphic development can be labor intensive, the additional workload on a developmental team can create problems if not addressed.

This presentation will discuss a methodology developed when a large number of graphics were needed in a short time, and little additional manpower was available. The ideas presented here focus on a design process that treats each piece of a graphic as potentially reusable. A spreadsheet management system for the "reusable graphic segment library" will be shared. The problems, limitations and efficiencies of this type of methodology will be presented.

A slide presentation that tracks the development of several graphics will be shown. The graphics developed for this project were used in CBIV, but the techniques are equally applicable to any graphic intensive project.

Virginia Tech
115 Hutcheson Hall
Blacksburg, VA 24061

*703-961-6941

COMPUTER ASSISTED CREATION AND ANALYSIS OF POPULATIONS AND SAMPLES

C. N. Moore*

Mississippi State University

The study of statistics may be viewed as the study of populations, variation, methods of data reduction and samples. Since populations in the "real world" are rarely seen and are usually only "assumed" in the classroom the concept of populations is vague. Tangible evidence of populations, samples, and variation is helpful in providing understanding of and appreciation for the theory and methods of statistics studied by students in various disciplines.

A modular program (SPAIS) for providing this tangible evidence was originally written in FORTRAN for batch processing on a large mainframe computer. It was used successfully in undergraduate and graduate classes in business statistics. Recently, a modified version of this program has been written in BASIC (SPAIS/PC) to permit interactive computing using either a mainframe or personal computer. In a series of Yes/No type decisions and inputs from the terminal, the user can create, analyze, and display various kinds and sizes of populations. Also, the user can easily select, analyze, and display random samples of various sizes from the populations. Empirical tests of the validity of well-known theorems (e.g., central limit theorem) can be performed.

*204 Cherry Lane
Starkville, MS 39759

601-323-5192

EVALUATING THE CBT COURSE:

QUESTIONNAIRES THROUGH CMI

Richard Reinhart*

Microtel Learning Services

We are entering an era of more and more computer-based training. Both the developer and the purchaser must be concerned with CBT courseware evaluation. How do you know it works? Cost and time contraints invite short cuts for this extremely important area. Establishing your evaluation goals is first and foremost. There are several areas here which may be confusing if not identified and segregated such as hardware/software, environment, student attitude and courseware. There are several evaluation methods which are available such as small group trials, "one-on-one" evaluations, pilot studies and quality control reviews. All of these methods can be supported by the tools of evaluation such as the questionnaire, interview, observation and pre and post course testing. The major evaluation advantage that CBT developers have over other instructional media is that of computer-managed instruction. Much of the tedious tasks of student observation and test statistics can be quickly and efficiently accomplished by an appropriate CMI system. Efficient use of available evaluation methods and tools is necessary to meet the demand of cost-effective, high quality CBT today.

*M1903-1600 Beach Avenue
Vancouver, B.C. B6G 1Y7

604-293-3680

THE PROJECT LIFE CYCLE FOR DEVELOPING CBT

Jerry Richard*

Texas Instruments

More and more data processing organizations find that structured analysis and structured programming techniques are used to speed up design and development processes and reduce maintenance costs. These techniques can do the same thing for computer-based training.

This presentation will discuss the merging of education methodologies and data processing methodologies to create the structured CBT methodologies used at Texas Instruments. Before beginning any development projects at Texas Instruments, a six step project life cycle is addressed covering the following areas:

1. Project definition and analysis
2. Design
3. Authoring
4. Validation
5. Implementation
6. Maintenance

This lecture will generally describe what each step accomplishes. The initial two steps (project definition and analysis and design) are covered specifically in the lecture and the audience will have handouts to pace through the lecture.

The project definition and analysis phase of the life cycle produces a target document (specification document) with appropriate strategy, goals and objectives defined.

The design phase of the project examines the use of the one entry/one exit rule for modules, limiting the size of modules, using hierarchy modules for menu design. These are just a few of the techniques which can assist in the development of quality CBT at universities and organizations.

*Texas Instruments
13777 N. Central
Suite 301, Keystone Park
Dallas, TX 75243

214-995-6582

THE INITIAL DEVELOPMENT OF SUBJECT CONTENT FOR INSTRUCTION BY COMPUTER TECHNOLOGY

Grosvenor C. Rust*

Montclair State College

There are a number of major considerations to which any developer of instructional materials for computer technology must give attention. One of them is the system of human sensory input. The senses are the interface between the brain and the computer. The nature of this interface is rather complex, probably more so than most program developers realize. This is dealt with for the next few minutes. The visual nature of the learner's experience with instruction from a microcomputer is examined, then the tactile, the auditory and certain other sensory paths allowing input to the mind.

Consideration of the concept of interaction with the computer is primary. The degree and kind is important. Our discussion expands on this and then goes on to discuss the links between the subject to be taught, the digital equipment, the modes available, the development of the content, and the manner of presentation.

Next comes the process of screening and filtering to avoid wasting time programming unsuitable material. Precise instructional objectives are absolutely necessary.

We consider the format and process of putting embryonic software on paper. There is a need to develop a model of what is to be programmed. Here consideration goes to the visual form for the overall sequencing of the instruction and the plan for interactive involvement. Consider the authoring systems' control over planning, the design and capacity of the computer and whatever peripherals are to be used. Conclusion with a summary of the salient factors.

*Montclair State College
Chapin Hall 111
Upper Montclair, NJ 07043
201-893-5183

MIDIS - A MICRO-COMPUTER BASED FLIGHT DECISION-TRAINING SYSTEM

A. Stokes,* C. Wickens, T. Davis

Institute of Aviation

Jensen (1982) points out that 80-85% of civil aircraft accidents can be attributed to "pilot error". His analysis of these errors shows that 50% of accidents are the result of faulty pilot judgment. Jensen writes "In spite of the statistics, attempts to teach...(judgment) are almost non-existent. It is but a slight overstatement to say that good pilot judgment is learned by the lucky and the cautious over many years of varied flying experiences". There is, however, recent evidence that pilots do indeed make better decisions after explicit pilot-judgment training courses (Buch and Diehl, 1984).

With this in mind, a microcomputer-based decision-training system has been developed, and takes the form of a stand-alone part-task trainer suitable for typical flying-school environments. The formal structure of MIDIS is that of a "graph traverse", in which the nodes of a branching structure are linked by transitional probabilities. Each node is a realistic in-flight situation referred to as a "scenario". The system models the chain of deteriorating circumstances which typifies the development of mishaps in complex multivariate decision-making contexts. A prime application of the system is to train students to break the chain. It has also been developed, however, as a research tool to facilitate the psychological study of pilots' decisions.

*Aviation Research Lab
Institute of Aviation
University of Illinois
Willard Airport
Savoy, IL 61874
217-333-3162

AIRBORNE RADIO-COMMUNICATION TRAINING - SPATIAL OR VERBAL PROBLEM?

A. Stokes,* T. Davis, M. Weller, C. Wickens,
B. Entwistle, R. Braune, A. Mane

Institute of Aviation

Most pilots have some initial difficulty with communications procedures and some have persistent problems. It is often assumed that the problem is predominantly verbal, i.e., the use of specialized jargon. In fact, aircraft transmissions are basically English in syntax and lexis. But they do tend to be degraded by a poor signal to noise ratio and are couched in particularly low redundancy English for the sake of brevity. Furthermore, student pilots devote most attentional resources to control and navigation. These factors may increase the difficulty an individual has in playing the "psycholinguistic guessing game" (Goodman, 1964) - the continuous process of making hypotheses about upcoming information from contextual cues in the message.

If so, the student must rely upon situational cues from the flight environment to anticipate the content of incoming radio messages. It may be the quality of a pilot's mental model of the three-dimensional airspace and the system of operations within it which primarily influences his ability to "key into" incoming calls and formulate appropriate messages of his own. This suggests that early flight training should emphasize the development of that model rather than the learning of phraseology alone.

Accordingly, a series of three CAI lessons were developed which blended audio presentation of radio messages with color graphics depicting airspace, controller jurisdictions, traffic areas, etc. Training using this approach was compared with conventional flight instruction techniques.

*Aviation Research Lab
Institute of Aviation
University of Illinois
Willard Airport
Savoy, IL 61874

217-333-3162

CBT PROJECT DECISION GUIDE

Adele Suchinsky*, David R. Losleben

Department of Defense

For anyone faced with the necessity of making decisions concerning the design/development of an instructional program on a computer, many factors need to be assessed. The three primary areas that need to be considered are hardware, software and courseware. These areas are complex and as varied as the approaches to the delivery of instruction.

To facilitate the decision making process, a checklist should be used to identify specific design/development questions/criteria prior to personnel selection, equipment purchases and contractual commitments. Once the items on the checklist have been satisfactorily answered, the difficult task of isolating a path from theory to practical delivery has been narrowed to an itemized scheme.

*NSA, ATTN: E21
9800 Savage Road
Ft. George G. Meade, MD 20755
301-859-4795

WRITING EFFECTIVE QUESTIONS FOR TRAINING DATA ENTRY OPERATORS

William Sweeters *

Ford Aerospace and Communications Corporation

This presentation describes the application of a cognitive model for training operators to perform data entry tasks.

Many data entry tasks require an operator to analyze data and fill-in fields on a computer screen. If done without aids, the operator must learn the location, name, and function of each field. This creates a set of six possible question formats. Operators may also need to learn specific information about particular fields. This creates a more complicated set of questions.

Frequently, data entry tasks require an operator to analyze a variety of situations. Job situations can be simulated by using a scenario which presents conditions and sets a task for the student.

The analysis to be presented will indicate that these question formats are likely to be most instructionally effective:

1. Do one or both of the following:
   (a) Given the location of a field, provide its name AND given the name of a field, provide its function.
   (b) Given the location of a field, provide its function.
2. Given the location of a field and a relevant situation, provide the appropriate value to be entered in the field.

*Ford Aerospace and Communications Corporation
7100 Standard Drive
Hanover, MD 21076

301-796-7800

FACTORS TO BE CONSIDERED IN EXPANDING AN EXISTING CBT SYSTEM - SHORT AND LONG RANGE PLANNING

Barbara J. Wright*

Federal Aviation Administration

FAA began to test the feasability of its current CBI system in 1974. In 1977 the first field test was conducted to study the application of CBI in a distributed setting using the CMI features that had just become available permitting centralized training management and quality control. Today there are over 500 CBT terminals being used in numerous FAA locations throughout the country and San Juan for Flight Inspection personnel, Engineers, Electronic Technicians and Air Traffic Controllers. Considerable thought/attention is presently being given to the factors that need to be considered for system expansion.

To optimize scarce resources for required proficiency training and for new Air Traffic employees, immediate expansions are needed. Other longer range system expansions are contemplated in the future because of a substantial increase in training requirements in a context of ever shrinking training staff and budget resources. For example, FAA is undergoing a massive multi-billion modernization program. With the already considerable training requirements these changes will further challenge the burgeoning training system.

This writer recommends that factors for system expansions be based on improved analysis of both short and long range training requirements, among other factors. In reality most organizations do not effectively plan beyond the short-term budget cycle.

*10771 Cordage Walk
Columbia, MD 21044

202-426-8865

COMPUTER COMPETENCY FOR SECONDARY TEACHERS

Lane F. Birkel*

Ball State University

The approach to computer competency for secondary teachers at Ball State University utilizes both a test-out procedure for demonstrating computer competence and a choice of courses which can be used to fulfill this requirement. The process has two stages, or levels. At the first level, the student may demonstrate a knowledge of the basic operation of the microcomputer and of the important computer terms and concepts. This may be done by taking an examination given by the microcomputer itself.

At the second level, the student must demonstrate a working knowledge of how the computer may be used in the instructional process. This will usually require some study in a special course given by the student's major department or by the Department of Secondary Education.

This latter course as a general approach to the use of microcomputers in instruction has been taught regularly for the past three years. It fulfills the requirements for both levels described previously. The course includes the operation of the microcomputer and peripherals, the evaluation and instructional use of commercial software, the creation of software, classroom applications in various subjects, and the use of the computer in managing instruction and processing information.

*Ball State University
Teachers College 802
Muncie, IN 46306

317-285-5475

CONCEPT PRESENTATION GENERATOR

Gerald Blumenfeld*

The University of Akron

The development of high quality tutorials and drills requires considerable effort and time. A major proportion of the time is consumed by designing the content and sequence of the instruction. Future authoring systems should replace some of the required instructional design and development efforts given an adequate knowledge base.

This presentation focuses on a program which will demonstrate some of the instructional programming aids useful in producing courseware that teaches concepts. The program uses a knowledge base provided by the teacher. This knowledge base requires the teacher to enter for each concept, information such as the concept's name, attributes, attribute values, and range of the values along the defining attribute. The program generated permits the student to see positive and negative instances, test instances and be tested. The program will also generate a tutorial sequence which uses commonly hypothesized guidelines for constructing an effective instructional sequence for concept learning.

Consideration will be given to various types of concepts and the problems of defining a specific concept well enough to permit the computer to generate the required instances and sequences.

*The University of Akron
Department of Educational Foundations
Akron, OH 44325

216-375-6956

WILL TECHNOLOGY IMPROVE EDUCATION IN THE UNITED STATES?

Alfred Bork*

University of California

Although computers have appeared rapidly in schools and universities, it is far from certain that they have, or will, improve the quality of education. At the present time the balance may be on the negative side; much of the existing educational software is of low quality, and so in typical classrooms the time spent using computers is wasted.

However, the modern interactive technology - computer, videodisc, Compact Disc Interactive, etc. - has the capability of greatly improving education. We can begin to help each student, on an individual basis in a learning environment much more active than our current learning environment. To do this we need extensive high quality technology-based learning material.

At least four issues must be considered: First, how is such material to be prepared; second, how do we convince the country or the world that this approach is vital to the national and international interests; third, what are the funding sources; and fourth, how is the new curriculum material to be introduced into the educational system?

We can improve education. But we must plan how we can proceed, and work toward a better future.

*University of California
Information and Computer Science
Educational Technology Center
University of California
Irvine, CA 92717

714-856-6945

COMPUTERIZING THE COMMUNITY COLLEGE CAMPUS

Linda A. Boyer*

North Shore Community College

While community colleges typically meet the needs of local communities by offering computer-related courses in programming, word processing, or lately, CAD/CAM, few advances have been made in integrating computers across the curriculum. At best, small pockets of interested faculty members in one or two disciplines (usually math and sciences) try "computerizing" their courses in haphazard and occasionally disastrous fashion. At North Shore Community College, institutional commitment to educational technology provides us with the opportunity to implement campus-wide computing strategies, policies and procedures.

Our initial focus in educational technology in 1984 was: 1) To provide support/services for individual faculty in identifying, reviewing and implementing existing CAI or productivity packages. 2) To develop institutional policies for integrating computers across the curricula. 3) To provide access to computing equipment.

To effect these goals, we've provided hardware/software support for half our faculty; identified courseware for each academic area/discipline; developed an on-line database of college-owned software; and conducted training and workshop sessions for both faculty and administrators. Additionally, we've opened CAI labs and software development and evaluation areas for faculty and student use; provided financial support for faculty to purchase courseware; and developed a plan for acquiring, storing, cataloging and distributing courseware on campus.

---

*North Shore Community College
Lynn, MA 01902
617-593-6332

COMPUTER GRAPHICS FOR KIDS

Barbara W. Fredette*, Grace W. Ferrero

University of Pittsburgh

Every catalog of computer software contains listings of graphics packages with glowing descriptions of how children can use them to create interesting masterpieces complete with vibrant color, animation and/or three-dimensional effects. The presenters--one an art and early childhood educator, the other a computer and elementary educator--have begun a systematic investigation of many of these packages. We began with LOGO, MacPaint, the Koala Pad and graphics tablet for the Apple microcomputers and are now working with Paint and Graphicraft for the Amiga. A general concern frames our investigation. Is there a concomitant cognitive gain entwined in the enticement of picture making fun? More specifically, the following criteria are the basis for the assessment of each graphics package: 1) Ease of use by elementary school children and their teachers; 2) quality of the visual elements permitted by the program; 3) number of different effects possible (e.g. animation, 3-D); 4) ability to save drawings as hard copy or on a disk; 5) opportunity to include text with graphics; 6) encouragement for children to produce original images.

Our purposes are twofold: 1) to prepare self-instructional modules for teachers attending courses in our Regional Computer Resource Center and 2) to be able to intelligently advise school personnel about the strengths and weaknesses of various graphics packages.

*University of Pittsburgh
4C21 Forbes Quad
Pittsburgh, PA 15260

412-624-1329

THE EFFECTS OF LEARNER CONTROL, ADVISEMENT AND PRIOR KNOWLEDGE ON ACHIEVEMENT AND INSTRUCTIONAL EVENT SELECTION IN CBI

Jane Marie Gleason, Wallace Hannum*

University of North Carolina

This study sought to determine 1) what instructional events should be included in CBI lessons; 2) who should determine the selection and ordering of instructional events, i.e. learner control or program control; 3) the relationship of prior knowledge and learner control on instructional event selection and achievement; and 4) the effects of pre-instruction advisement on instructional event selection and achievement.

One hundred and six seventh grade subjects were randomly assigned to five treatment groups for a CBI lesson on probability. One group received learner control and pre-instruction advisement regarding instructional events; one group had learner control without advisement. Three groups had program control over instructional events. One group had all instructional events (high instructional support), one group had a reduced set of four instructional events (moderate support), and one group had a partial set of three instructional events (low support). The subjects in the study completed a pretest, a post-test and a one week retention test.

The results are discussed in terms of the prior research and theory. Suggestions are made for the design of subsequent research studies and for the design of CBI lessons. Handouts and visuals will be included in the presentation.

*University of North Carolina at Chapel Hill
School of Education
Chapel Hill, NC 27514

919-962-3791

THREE TIERED APPROACH TO INSTITUTIONAL COMPUTING

Floyd M. Grimm III*, Richard V. Miller

Harford Community College

Today, it is recognized that the use of computers on the community college level is a necessity and not an expensive luxury. However, many institutions have expended large sums of institutional funds to purchase hardware and software only to find that both are unused. The bottom line--funds were misappropriated and institution members became more apprehensive about computers. Institutions research, investigate, plan and finally implement a coordinated and integrated approach to institutional computing.

For the past three years, Harford Community College has been involved in a "Three Tiered Approach" to an institutional (administrative and academic) computing program. The three tiers (levels) of computer needs at the college include: 1) administrative; 2) data processing (career) instructional programs and 3) the general computer needs of all other instructional programs on campus. Each of these tiers has some commonality and each requires some unique hardware/software items.

In order to meet these needs, the college identified five steps which needed to be accomplished: 1) determine the software needs of each tier; 2) acquire the hardware; 3) orient and develop faculty and staff; 4) implement the systems and/or instructional program(s) and finally, 5) evaluate results and take corrective actions if necessary.

Harford's response to each tier's needs was thoughtful, systematic and timely. Throughout the planning and implementation of each step of the process, both the personal and educational/developmental needs of students, faculty and staff were given high priority. Harford's results? The breadth and sophistication of institutional computing on campus today is on the technological and educational cutting edge!

*Harford Community College
401 Thomas Run Road
Bel Air, MD 21014
301-836-4372

DESIGNING EFFECTIVE LEARNER-COURSEWARE INTERACTIONS:

RESEARCH BASED RECOMMENDED ACTIONS

Wallace H. Hannum, David H. Jonassen*

University of North Carolina

To some extent, the demonstrated effectiveness of computer-based instruction is often thought to be a function of the nature of the interaction of learners with the courseware that CBI allows. There is scant evidence that the hardware itself accounts for the positive findings (Clark, 1983). If there are no hardware effects and if the interaction is an important aspect contributing to the effectiveness of CBI, then the design of more effective interactions should result in more effective CBI lessons. This presentation assumes such a premise.

Based upon recent research findings related to learner-courseware interactions (Hannum and Gleason, 1986) we will make a series of recommendations about how to design user-courseware interactions. These suggestions will be specific and prescriptive. They will not be hardware or software (computer language, authoring system) specific.

After attending this presentation, participants will be able to apply principles of learner-courseware interactions to designing effective interactions for various learning situations, focusing on how to design interactions that stimulate meaningful cognitive processing on the part of the learners. The presentation will include recommended prescriptions for interactions and examples of appropriate interactions. These recommendations will relate to four aspects of the user-courseware design -- learner orientation, instructional stimuli, learner responses and feedback. Handouts that summarize the suggestions provide examples and relate the suggestions to the relevant research evidence will be provided.

*University of North Carolina Greensboro
School of Education
Greensboro, NC 24412-5001

PROGRESS REPORT

CAI FOR PASCAL IN THE HUNTINGTON PROJECT TRADITION

Jesse M. Heines*

University of Lowell

Faculty and students at The University of Lowell have begun developing instructional modules to help first-year programming students with difficult concepts in Pascal. Such concepts include parameter passing, pointers and their use in linked lists, binary search trees, recursion and sorting algorithms.

These modules supplement, rather than replace, traditional teaching methods. They are designed for use both in the classroom and by students working on their own. One type of module presents a short review of a concept followed by an interactive session in which students see Pascal code and a visual representation of its effects. Each line of code is highlighted as it is processed, with the corresponding effects on variable values or memory allocations shown in another area of the screen. A second type of module allows students to enter actual Pascal statements, parses them, provides detailed explanatory messages for incorrect statements, and gives "clues" to solving a "murder mystery" for correct statements.

All modules are being developed in a format similar to that employed by the Huntington Project in the early 1970s. That is, each module includes a comprehensive student guide and a matching instructor guide to supplement the interactive program. The development of high quality documentation is an integral part of this project and is intended to increase the usability of the modules at other institutions where the sequence and focus of instruction may vary significantly.

---

*University of Lowell
Department of Computer Science
Lowell, MA 01854

SCIENCE INSTRUCTION ON GLOBAL ELECTRONIC NETWORKS

James A. Levin, Michael Waugh*

University of Illinois

Real world science is conducted in a global context, with collaboration among colleagues around the world a common occurrence. With the spread of microcomputers into science classrooms and the availability of electronic networks, it is now possible to conduct science instruction in such global contexts as well.

In this presentation we will discuss ongoing research into the ways in which such networks can be used in science teaching and learning. These networks allow students to participate in distributed science problem solving activities by providing dynamic support so that students can gather and analyze data, and jointly create project reports with students from other parts of the world. We will discuss two specific examples in detail. In one project, students from around the world have been engaged in a study of the problems associated with shortages of water. In another project, teachers and students will participate in "tele-science fairs" where the results of joint science projects will be reported and evaluated electronically. We conclude with a discussion of the potential of such networks for improving science teaching and learning.

*University of Illinois, Urbana-Champaign
386 Education Building
1301 S. Sixth Street
Champaign, IL 61821

217-333-0227

LOGO: A LANGUAGE FOR ALL GRADES

Robert A. Lucking* - Old Dominion University

Cherie Harrell - Tidewater Community College

LOGO is rightfully receiving more and more attention from elementary teachers. These teachers are discovering that LOGO's turtle graphics represents a marriage of computer applications and mathematics which allows youngsters to explore arithmetic and geometric concepts in an exploratory environment. However, LOGO's list processing ability also allows it to be used in the study of language arts. Students can use LOGO to engage in such diverse activities as identifying parts of speech to generating poetry.

Teachers of older students are discovering that LOGO is also useful as a tool in other disciplines such as science and in the creation of CAI materials. New peripherals and software allow LOGO to be used in laboratory situations for a wide range of experiments of natural phenomenon. Additionally, LOGO allows for the simple combination of graphics and text and is, therefore, especially well suited for the creation of drills or tutorials in any subject.

High school teachers of languages are also finding that LOGO provides the appropriate introduction to PASCAL. Because LOGO is a procedural language, students who are comfortable with LOGO can more quickly acquire a command of the syntax of PASCAL and the principles of structured programming. In summary, LOGO is becoming the chosen language of K-12 education.

*Old Dominion University
Department of C&I
Norfolk, VA 23508

804-440-3283

COMPUTERIZED INSTRUCTION AND DISTANCE EDUCATION: PENN STATE'S INNOVATIVE SOLUTION FOR THE ISOLATED ADULT ENROLLED IN INTRODUCTION TO STATISTICS IN EDUCATIONAL RESEARCH

Marilynne W. Mathias*

Penn State University

In general, distance learners are hampered by delayed feedback on their work. This coupled with the complexities and anxieties associated with a college level statistics course, led to a concern over a potentially high student drop-out rate. Thus, when a new independent learning course was being developed, a decision was made to include a microcomputer component as an aid to reduce the isolation of the learner at a distance.

While a computer-assisted instruction (CAI) module would appear a logical format, the disk was designed to be supplemental to a comprehensive textbook and study-guide. As such, it evolved into a demonstration, practice, diagnostic, remedial disk with a primary objective of supplying immediate tutor-marked feedback. The developed disk became a substitute for the teacher-student activities which typically occur in the course when taught in a resident-instruction format. By not initially categorizing the disk as solely drill, tutorial, or simulation, this supplementary instructional tool contains a variety of support programs - each reflecting a critical unit of instruction.

Practice is an important step for a student to acquire confidence and competence. Distance education students cannot interrupt a faculty member demonstrating a problem at the chalkboard. The computer disk encourages student computational skills by requiring calculation, but rather than only giving a final correct response, it provides student checks at various strategic points. This process permits pinpointing error points and provides student motivation to continue working laborious problems. Because of these features, the disk is also available to students enrolled in the traditional University Park course.

The presentation will demonstrate the software and present information on the resulting attrition rate.

*Penn State University
303 Wagner Building
University Park, PA 16802

814-863-2271

EFFECTIVE TEACHING IN CLASSROOM GROUPS:

A COMPUTER-BASED ALTERNATIVE

Steven L. Robinson*

Amherst H. Wilder Foundation

The proposed topic presentation will discuss research related to and demonstrate an innovative, computer-based, classroom communication system. Most technology-based instructional applications focus on delivering instruction to individuals in the absence of a teacher.

The function of the system to be discussed is to keep the teacher at the center of instruction and to increase teacher control over critical elements of effective instruction in classroom groups. These elements are: classroom management, student participation and instructional interaction. The system, called the Discourse Educational Communication System, permits all students in a classroom to respond on any given occasion, under the direct scrutiny of the teacher. Full sentence responses are typed on individual keyboards with displays which are connected to a single computer used by the teacher. Responses are monitored by the teacher via the computer monitor display. Teachers also use the system to control the presentation of slide, videotape, videodisc and text material during instruction.

The Wilder Foundation has spent the last two years investigating the effects of this system in school-based classroom settings. A summary of findings from this research will accompany the demonstration.

*Amherst H. Wilder Foundation
919 Lafond
St. Paul, MN 55104

612-642-4022

BIBBITS AND GRIBBITS: AN ADVENTURE IN LOGICAL THINKING AND PROBLEM-SOLVING WITH PRIMARY CHILDREN

Charlotte Scherer*

Bowling Green State University

First and second graders were taught the seven logical thinking and problem-solving games on Addison-Wesley The Learning Company's Moptown Parade program. With university student assistance, the children were introduced to the Moppet characters and their various traits: that of being Bibbit or Gribbit, tall or short, red or blue and fat or thin. During each of six one-hour sessions the children learned and played the games and completed some "off computer" worksheets that extend and make applications of the skills learned.

Goals for the six sessions were several: 1) to observe the children's reactions to the program; 2) to determine whether the children would become adept at the skills of defining and identifying traits, using problem-solving, making twins and opposites, finding differences, identifying patterns and using deduction to decide members of a group; 3) to determine whether the children could transfer the skills to reading and math; and 4) to determine whether the children enjoyed working with computers to learn educational skills through a game approach. During the sixth session, the children were tested using paper and pencil exercises similar to those on the worksheets and given a questionnaire to evaluate the accomplishment of the goals. Procedures, materials, results and recommendations will be discussed during the presentation.

*215 West Washington
Bowling Green, OH 43402
419-372-7392

PHOENIX AS A COST EFFECTIVE AND FLEXIBLE MAINFRAME CAI SYSTEM AT THE JUNIOR COLLEGE LEVEL

Gary Vaughn*

University of Cincinnati

The presentation will examine the successful application of the Phoenix CBT system to students, mainly on a developmental level at a two-year junior college. The CAI Lab in the University College of the University of Cincinnati offers forty one programs to students who use the Lab in three ways: 1) by dropping in on their own initiative; 2) by being referred by individual instructors; 3) by registering for any of the three accredited courses held solely in the Lab (CAI Basic English Skills Lab, CAI Mathematics Lab, CAI Introduction to General Statistics). The latter option is open to all students at the University and generates income from tuition that subsidizes the first two options.

Graphs will be presented that show the yearly income produced by the accredited courses more than offsets the yearly operating expenses of the CAI Lab, including maintenance and repairs, equipment, salaries, and the small percentage of time (under 1%) that Phoenix uses on the University's Amdahl mainframe computer. Thus, the proper set-up of a CAI lab for college students can not only be cost effective but actually income producing.

Furthermore, Phoenix has the ability to adapt to the growing popularity of personal computers over mainframe-based CAI. Phoenix programs can be easily downloaded to IBM personal computers. In short, the CAI lab that uses Phoenix has both the power and central control of a mainframe and the flexibility of a microcomputer.

*University of Cincinnati
M.L. #203
Cincinnait, OH 45221

513-475-6859

THE EFFECT OF STUDENT FEEDBACK ON ACHIEVEMENT

Elizabeth West*

University of Illinois

The purpose of this study is to evaluate the influence of the timing and type of feedback used in a CAI tutorial lesson. This presentation will summarize the literature in this area and present the results of a study done with a population of university students on the PLATO computer system.

This study will present students with a short tutorial lesson chosen randomly from one of two variations of this lesson. One form of the lesson will require students to immediately confirm the correct response to any item answered incorrectly. The other form of the lesson will present the correct answer in response to an incorrectly answered item and will not require student confirmation of the correct answer. The lessons will be identical in all other respects. A post test will be given at the end of the tutorial. Post test scores will be analyzed to ascertain whether the requirement of immediate student confirmation of the correct response has any influence on student achievement.

*University of Illinois at Urbana-Champaign
395 Education Building
1310 S. Sixth Street
Champaign, IL 61820

217-333-0227

INTEGRATION OF COMPUTER ASSISTED INSTRUCTION
WITHIN A BACCALAUREATE NURSING CURRICULUM

Jean Arnold*, Elsie E. Gulick

Rutgers The State University

A large baccalaureate nursing program was able to develop a microcomputer nursing laboratory and initiate computer assisted instruction partially due to grant funding. Since this was a major change within the undergraduate curriculum it was critical to develop a design for implementation and an evaluation plan. The project was coordinated by two nursing faculty who had specialized roles. One member served as implementation coordinator and focused on the design of computer laboratory, including selection of hardware and software as well as day-to-day operations with faculty and students. The other member focused on the research component which included measures of attitudes, learning style and evaluation of software by both faculty and students.

The implementation of CAI began with development of a microcomputer laboratory consisting of 10 microcomputers. Management problems that arose included access due to 1:47 student machine ratio, need for faculty training, lack of technical access and availability of only one copy of computer assisted instruction software for use by a large number of students.

Evaluation of student attitudes toward CAI and the effectiveness of CAI as a teaching-learning strategy from the perspective of the student and faculty will be completed at the end of the Spring 1986 semester.

*175 Atlantic Drive
Parsippany, NJ 07054
201-648-5293

INTERACTIVE VIDEO DISCS AS TUTORS FOR VISUAL LEARNING IN MEDICAL EDUCATION

Terry S. Field*

Boston University School of Medicine

A large portion of the knowledge which undergraduate medical students must acquire in such courses as anatomy, histology, pathology, radiology, ophthalmology and surgery is based on visual material. Video discs, with their capacity to hold and instantly access 54,000 still frames; and computers, with their increasing ability to both control disc players and easily interact with users, hold great promise as major facilitators of this learning process. A number of medical educators are experimenting with this approach. The main drawback has been the enormous cost of traditional, commercial modes of producing and programming interactive video materials.

This presentation will review a variety of special purpose and "generic" visual data bases on disc intended for use within medical education. The focus will be on two interactive video projects being developed in-house at a medical school to support the visual components of courses in histology and pathology. It will address issues related to optional design approaches for accessing visuals on a disc, the use of graphics overlaid on photographic images, and some production and management approaches which drastically reduced costs of these projects.

*Boston University School of Medicine
80 E. Concord Street
Boston, MA 02118

617-638-5147

DEVELOPMENT OF INTERACTIVE VIDEODISC PROGRAMS FOR BASIC MEDICAL SCIENCE EDUCATION

Ronald P. Jensh, F. Scott Beadenkopf*

Jefferson Medical College

Computer assisted instruction has been of limited value in those basic medical sciences in which the study of morphology (shape and texture) plays a major role, such as gross anatomy, histology, neuroanatomy, pathology and microbiology. Inexpensive computer graphics systems have been unable to provide the necessary resolution and color definition, and coordination of computers with random access slide carousels or videotape machines has been cumbersome. With the development of laser videodisc technology and subsequent inexpensive computer interfaces, it has become possible for educators to augment computer generated text and graphics with near photographic quality images.

The objectives of the present project were to: a) determine the type of videodisc which would serve the needs of many different faculty, b) produce a pilot disc for faculty use, and c) create a program which could easily be used by faculty with limited computer expertise. It was determined that a videodisc generic library of material from which a multitude of programs could be developed would best serve the needs of faculty. A pilot disc was produced containing about 1,200 slides and a variety of audio and visual segments. Programs have been written to facilitate the development of instructional applications, including a simple videodisc indexing utility; and programs have been written for student use. The process of creating the videodisc, developing programs, student use, and student reactions will be discussed; and the videodisc will be demonstrated.

*Jefferson Medical College
1025 Walnut Street, Room 119
Philadelphia, PA 19107

215-928-8907

DEVELOPING CBE TO REPLACE A DECEASED PROGRAM

Victoria Moore*

Rush University

Test item banking has been performed for as many years as written exams have been administered. In the "old days" (pre-computer) whole tests were filed in cabinets. Today's technology allows instructors to store individual questions in a systematic fashion on disc. Future exams then may be generated from the bank once a sufficient number of questions have been entered.

The nursing pharmacology faculty and the Office of Computer Based Education at Rush University spent the 1985-86 school year developing a bank that provided weekly quizzes and tutorial sessions to be delivered and analyzed on-line.

Since 1979, Rush University's Office of Computer Based Education has been working with faculty members to automate components of their courses that have been difficult to teach or require extensive one-on-one interactions. The motivator that proved to be most effective in getting the College of Nursing to incorporate computers as an integral component in their curriculum was the cutback of teaching assistants. The pharmacology course had previously offered weekly quiz/tutorial sessions, but due to budget constraints, that part of the course had to be eliminated. It was a great opportunity to demonstrate how CAI might fill the void in services offered to students taking pharmacology.

No faculty were available in the initial stages to develop a hierarchical structure so questions were entered in a chronological format. Later, a subject matter hierarchy was implemented and questions transferred to appropriate locations. The subject matter experts attempted every conceivable short-cut and found that the guidelines recommended by the Office of Computer Based Education were developed with considerable thought. Some of the pitfalls and success of the development and implementation will be presented in this session.

*Rush University
Office of CBE
600 South Paulina
Chicago, IL 60612

312-942-6557

PRODUCTIVE GAMING

Victoria Moore*

Rush University

Many computer users seem to be entranced by electronic games. Medical students, despite school commitments, are no exception. In fact, students use games as both a means of recreation and as an educational resource. In an effort to meet both needs, a game is being developed at Rush University for medical students preparing for national boards.

The game is based on a self-generating maze format with player movement being determined by the student's ability to answer questions correctly. The lowest level of the maze deals with the basic sciences, the second level with clinical information. A player must demonstrate a mastery of all the basic sciences before gaining access to the higher level clinical subjects.

The successful movement around a gameboard is not sufficient to hold a player's interest, so the next phase of development required building a reward system -- game points, player status, acquisitons and a recordkeeping system so that players may exit and resume a game midpoint. There was never any question about maintaining a "hall of fame" to store and display best scores. It is a favorite feature of most computer games on PLATO.

The game is still in development, but it is anticipated that within the 1986-87 academic year the game driver will have been completed and the bank of medical questions entered. A provision for attaching datasets to store questions separately from the game program will allow any subject matter questions to be used.

*Rush University
Office of CBE
600 South Paulina
Chicago, IL 60612

ROUTINE HEALTH SCREENING

Daniel K. O'Neil*, Alan G. Adler

Jefferson Medical College

Preventive health care has been ignored in both the traditional curricula of medical schools and of residency training programs. The few studies of preventive health care delivered by residents during training demonstrated performances far below the ideal recommended by the experts. Among patients meeting accepted criteria for the procedures 5 to 19 percent received pneumococcal vaccinations; 4 to 30 percent received influenza vaccinations; 29 to 97 percent received breast exams.

In another study, practicing internists not only ordered many tests that were not recommended, but also they omitted many tests that were suggested. To fill this educational void, we developed a teaching system on Routine Health Screening using the PILOT (Program Inquiry Learning or Teaching) language. This program includes interactive teaching modules on mammography, sigmoidoscopy, cardiac studies, tonometry, immunizations, pelvic exam with pap smear, lab tests, stool for occult blood, and history and physical. Presently, this interactive program is used by medical students and residents in our teaching service and clinics.

*Jefferson Medical College
Box 711 Jefferson Alumni Hall
1000 Locust Street
Philadelphia, PA 19107
215-336=6283

BUILDING EXPERT SYSTEMS

A POWERFUL INSTRUCTIONAL TECHNIQUE

Stanley R. Trollip*, Renate C. Lippert

University of Minnesota

Expert systems are rapidly becoming common tools in many different environments. Some work has also been done in using expert systems or their knowledge bases as an instructional tool. However, the effort and cost of developing such systems is great. Little explored is the use of building simple knowledge bases and associated expert systems as a means of teaching difficult topics. By asking students to become knowledge engineers forces them to isolate essential elements and to focus on important relationships. Implementation heightens motivation and allows automatic checking for consistency of rules.

This paper discusses work done at the University of Minnesota in this area, focusing particularly on issues such as domains of reference and educational levels of applicability.

*University of Minnesota
148 Peik Hall
Minneapolis, MN 55455

612-373-2660

THE DEVELOPMENT AND VALIDATION OF ATTITUDE SCALES AS COVARIATES IN CBI RESEARCH

Charles S. White*

George Mason University

The effectiveness of computer-based instruction may be determined in part by the attitudes students bring to the computer experience. Studies can control for the effects of such intervening variables statistically be measuring subjects' entry attitudes and using the scores as covariates, helping to insure that differences between treatment groups derive from "real" differences in the materials or methods being examined, rather than from attitudinal differences. This presentation focuses on the development of two attitude measures for use as covariates in CBI research: a Computer Attitude Scale (CAS) and a Problem-Solving Attitutes (PSA) scale. The influence of the former attitude domain is supported by the CBI research literature, while the latter figures prominently in the problem-solving literature. Since a significant focus of CBI research is on the efficacy of computer-based instruction for problem-solving skill development, the PSA scale is of particular interest.

The presentation will describe how the scales were constructed and how they were validated. Reliability and content and construct validity data will be presented, and the performance of the scales in a recent research study will be described. Copies of the two scales will be made available to participants.

*George Mason University
Education Department
4400 University Drive
Fairfax, VA 22030

703-323-2421

Evaluation of Diagnostic Tests

Scott Worman*

Thomas Jefferson University Hospital

There has been a dramatic rise in the cost of health care in recent years. Laboratory tests which are used to aid in the diagnosis of disease comprise a large portion of these expenditures. Unfortunately, many physicians are ordering tests in an inappropriate fashion because they do not understand how to correctly judge the utility of a laboratory test within a clinical framework.

We have developed a interactive tutorial computer program which is designed to help a student learn to evaluate a particular laboratory test. Designed for use by medical students and physicians, the program teaches the concepts of accuracy, precision, sensitivity, specificity and predictive value. The program uses a series of multiple choice and free response type questions and requires the student to make several calculations to demonstrate his or her knowledge.

The program is written using the PILOT language and is designed to be used on an IBM-PC with color monitor. By using a multi-window screen design, the program is able to show the questions, the response to the answer and the correct calculations simultaneously.

*Thomas Jefferson University Hospital
Department of Pediatrics
700 College
Philadelphia, PA 19107

215-928-6525

USING A SIMULATION TO TEACH PSYCHO-SOCIO-ECONOMIC ASPECT OF CLINICAL MEDICINE

D. P. Yens*, J. T. Peters, M. C. Fahs, J. Lowe

The Mount Sinai School of Medicine

The Mount Sinai School of Medicine has received a grant from the National Fund for Medical Education to develop a computer-based system to introduce medical students to the psychological, social, and economic factor that can affect clinical decision making. Three modules will be developed; the first has been completed.

The first module makes the first instructional use of POLICY, a judgment analysis program that has been used for clarifying issues in policy decisions, and also uses TENCORE, a CAI author lanaguage. These programs are used to simulate different "cases" in which the student makes decisions about whether or not a person will seek medical care based on specific economic, social, psychological and medical factors. The weight that the student places on these factors and the relationship between his/her responses and the levels of the factors are displayed to the student and compared with those of experts so that the student can assess and modify his/her judgments. Collateral reading materials are provided with the simulation problems to provide a basis for the judgments.

Future modules will incorporate decision analysis techniques and clinical problems into the learning process.

*Mount Sinai School of Medicine, Anbg. 5-16
New York, NY 10029

212-650-6691

THE COMPUTER AS A TUTOR IN DIETETIC EDUCATION

Leta P. Aljadir*

University of Delaware

Thirteen computer-based lessons for senior dietetics courses have been developed in recognition of the need for individualized instruction in these courses. There is an expanding body of knowledge which requires more efficient use of classroom instruction. In addition, prerequisite science courses are generally taken as many as four semesters before these dietetics courses. The computer-based lessons integrate material from prerequisite science courses with basic dietetics concepts, and provide the students with the opportunity for self-paced learning.

Topics of these lessons include: metabolism of carbohydrates, proteins and fats; metabolic aspects of weight control; nutritional management of diabetes mellitus.

The choice of the computer as the medium for these self-instructional modules was based primarily on the capability for interaction and immediate feedback. These are not generally available in other systems. Additionally, the computer-based instruction system at the University of Delaware (PLATO R) provides the capability to monitor students' progress. The system is also available at several teaching institutions around the country. This provides an opportunity for sharing materials which are developmentally cost- and labor-intensive.

Ten of these lessons have been student tested. Student feedback has been highly positive. Several students have requested the use/ development of more of such lessons.

*University of Delaware
Department of Food Science and Human Nutrition
Allison Hall
Newark, DE 19716
302-451-2139

THE EFFECTIVE CLASSROOM USE OF MICROCOMPUTERS IN MERCHANDISE PLANNING

Holly E. Bastow-Shoop*

North Dakota State University

This presentation outlines the use over a two year period of a two-stage method for educating fashion merchandising students in the effective use of the microcomputer in retailing.

The first phase of the project was to develop a software program for a spreadsheet merchandise budget assignment. The budget was developed through a series of computations which provided figures for planned sales, sales percentage goals, stock to sales ratios, planned purchases and markdowns. Students worked through the computations by hand, taking an average of four to six hours. The same figures were punched into the computer and immediate results were provided. Students were convinced of the time-saving factor in using the microcomputer for developing and revising merchandise budgets weekly.

The second phase of the project was to utilize the same spreadsheet merchandise budget software to develop the students' own merchandise budget for the store of their imaginations. Student teams developed a scenario which outlined the type of store they wanted to eventually own or operate. They were to provide a detailed description of the management system, customer profile, price lines, merchandise assortments, physical layout and qualitative merchandise plans for a six-month period. Once the plan was developed they had to analyze their plans and explain their actions. The report included an analysis and evaluation of their scenario as well as alternatives and directions for future plans.

*North Dakota State University
Department of Textiles and Clothing
Fargo, ND 58105
701-237-7352

AN INTERDISCIPLINARY APPROACH TO UTILIZING COMPUTERS IN HOME ECONOMICS

J. Cheyer, B. Quinby*

Manchester High School

The department heads of computer science and home economics combined talents to create and institute a project to implement the use of computers in the home economics department. Together they wrote a grant that was successful in obtaining funding to pay for computers and software for use in a child development course at Manchester High School.

Problems encountered included lost data, training of beginning students and teachers and interpretation of instructions in the manual for use in an educational setting. Benefits derived have included increased public relations for the departments both locally and nationally, increased attendance in classes, motivation to learn and improvement in the quality of research papers required in the course.

*42 Peterson Road
Vernon, CT 06066
203-647-3521

KNOW YOUR MAN-MADE FIBERS

Nadine Hackler* - University of Florida

Linda Heaton - University of Kentucky

Know Your Man-Made Fibers is a practice computer program to educate individuals, especially youth, about man-made fibers. The program is designed for Apple IIe microcomputers and IBM/PC microcomputers (and their compatibles).

Textile products in today's world often confuse individuals both young and old. Many people lack the knowledge about textile fibers necessary to make appropriate decisions in selecting apparel and home furnishings. This computer program will aid people of all ages who want to learn more about textiles. The program focuses on the characteristics, uses and care of man-made fibers.

Accompanying the program is a pre- and post-test to evaluate the degree of learning. These instruments were used in the pilot testing of the computer program. The pre-test should be administered prior to the introduction of the study on man-made fibers. The post-test should be administered after the program is completed.

A dictionary of the terms used in the program appears as a part of the screen menu. To enhance learning, program participants should begin by experimenting with the dictionary and become familiar with the textile terms before using other parts of the program.

The computer program should be an integral part of the educational unit of man-made fibers. Evaluations from two pilot tests indicated the program would serve as a good tutorial method for students (junior high through college), as well as for volunteer leaders in Extension.

*University of Florida
Department of Home Economics
3002 McCarthy Hall
Gainesville, FL 32611

904-392-1945

INTEGRATION OF CAD/CAM AND
THE INTERIOR DESIGN PROGRAM

Susan Harshbarger*

Illinois State University

Computer literacy is necessary to keep students abreast of technological advancements in the interior design field. To help solve this problem, this educator has developed through an Illinois State University grant instructional materials, skills and expertise by increasing the participant's skills and understanding of CAD/CAM.

The presentation examines the integration of computer technology at various levels of the interior design production environment. Furthermore, an analysis will be done on the components of a CAD/CAM system, including an understanding of computer graphics, simulation, resource planning and group technology.

The main objective of this presentation is to communicate the integration of computer technology into the interior design curriculum. This objective is met through instruction, hand-outs and visual aids. The majority of this information was derived from personal experience with a computer and direct communication through site visits to various interior design firms, utilizing the technologies of CAD/CAM.

CAD/CAM is a technology that has widespread applications with the potential to be transferred to several disciplines, thus many individuals could effectively utilize the instructional material and expertise gained as a result of their participation in this presentation.

*62 Glenn Drive
White Heath, IL 61884

309-438-5909

INTRODUCTION OF TELECOMMUNICATIONS INTO VOCATIONAL HOME ECONOMICS PROGRAMS

Ann B. Kapp*

Northeast Louisiana University

Communications is an important part of the productivity gained with the personal computer. Instead of emphasizing computer literacy, this program introduced students and teachers to the use of computers for accessing information from many sources for use in the classroom and the home environment.

A network of TANDY 1000 microcomputers, located at NLU and in high school home economics departments, was established. Students and faculty used electronic mail between the university and the participating secondary programs.

In-service training was an integral part. After an introductory workshop, an NLU faculty member visited each school three times: the first, to deliver the system; the second, to teach all regular classes and an adult class one day; and the third, to evaluate and remove the system.

Objectives were for participants to be able to 1) describe the process of telecommunications and requirements for both hardware and software, 2) use the computer for accessing information from distant sources, 3) list and describe the kinds of information that can be secured, 4) evaluate the usefulness of the microcomputer in a home as an information provider and 5) feel comfortable in using a computer to accomplish a task. Additionally, the student teacher, college supervisor, and cooperating teacher participants would be able to 6) evaluate telecommunications as a supervisory technique.

---

*Northeast Louisiana University
Department of Home Economics
Monroe, LA 71209-0560

318-342-2155

DEVELOPMENT OF A TIME MANAGEMENT COMPUTER PROGRAM

Judith S. Kline, Emily S. Wiggins*

Clemson University

The objective of this project was to design a contemporary wedding package to help a bride-to-be plan her wedding. The components of the package were determined by the objective. However, in the Cooperative Extension Service one message is delivered to the clientele in many different media. This package contains printed material and planning sheets as well as a computer spread sheet for wedding record keeping.

Additionally, an interactive program allows the production of a personalized task time table for both the bride and groom. Certain tasks should be performed six months ahead of the wedding. What if there is only three months left? This program takes care of that in a unique way.

Educational radio and TV programs with the same information and even an interactive video are planned.

We envisioned the computer part of the project being used in shopping malls in early spring across the state just before June, the wedding month.

The project consisted of these steps: 1) searching to see if a similar wedding project was available elsewhere, 2) gathering information, 3) critiquing spread sheets and data base systems, 4) choosing one or the other, 5) explaining the project to a programmer, 6) editing, 7) packaging, 8) printing, 9) testing with brides and their mothers, and finally 10) marketing the package to Extension clientele both in and out of state.

*Clemson University
243 P & AS Building
Clemson, SC 29634-0312

803-656-3288

SPREAD SHEETS FACILITATE THE TEACHING OF FASHION MERCHANDISING

Grovalynn Sisler*

Oklahoma State University

Although in 1984 a research study showed that the attitutdes of home economics faculty toward computers are generally favorable, home economics teachers have sometimes been slow to incorporate computer applications into their classrooms. One of the reasons for this is the lack of appropriate software. Very little high quality commercial software specifically designed for home economics is available. The time and/or money required for the development of custom software has not been readily available. One solution may be to adapt commercially available software for specific classroom needs.

Fashion merchandising courses require a number of mathematical calculations in planning sales and merchandising assortments, and in determining open-to-buy. Commercially available spread sheet programs are designed to facilitate the calculations and formatting of reports such as those used in the retailing industry.

This presentation will focus on how to use a spread sheet program to minimize teacher preparation time and maximize student learning in fashion merchandising classes.

*Oklahoma State University
College of Home Economics
Clothing, Textiles & Merchandising Dept.
Stillwater, OK 74078

405-624-5034

SHOPPING FOR A REFRIGERATOR
VIA THE COMPUTER

Louise Wesswick*, Edith A. Miller

University of Wyoming

This is a slide presentation including 28 slides designed to demonstrate the way in which a computer program can be used to assist the consumer in acquiring pre-shopping information related to the selection of a refrigerator. Each slide is an exact duplicate of what would appear on the TV monitor if a consumer were to run this program on the appropriate computer. The graphics are designed to animate and enhance the program; the interaction provides the user with an opportunity to "answer" the computer, and the feedback is immediate to capture the interest of the user.

Field testing of the program is designed to 1) discover whether the program is "user" friendly or not, 2) determine whether or not the program is too short, too long, or just about right in length, 3) evaluate the content for interest, ease of reading and ease of comprehension, 4) examine the possibility that a computer program can leave the user with a vivid picture in his mind regarding the refrigerator or any other appliance, and 5) to learn whether or not consumers enjoy using computers as shopping aids. Further input by users of the program include what they liked best about the program, what they liked least, and whether or not they would recommend this program to others. Field testing results could be used to improve the design of computer programs so that those who have never used computers before will be attracted to their potential.

*2200 Garfield Avenue
Laramie, WY 82070

307-766-2211

THE COMPUTER'S CONTRIBUTION TO SOLVING SOME PROBLEMS IN SECOND LANGUAGE READING TASKS

D. Boatman,* C. Cieri, N. O'Connor

University of Pennsylvania

This paper will discuss two computer tools which simplify the second language reading task. Differences in individual learning strategies and knowledge bases motivate the implementation of tools which facilitate lexical and grammatical aid to the reader, increase retention and productive use of new structures and allow for individual reading styles.

The student reading a second language faces new lexical items and grammatical constructions. Various types of reference sources provide such help, but too often in a limited format which fails to allow for individual strategies in language learning. The second language reading tool we present allows the student to use his/her own strategy in accessing help.

Completing a reading task gives no assurance that one will be able to retain, or use productively, the novel items and constructions from the text. Reading comprehension tests also limit the student's decisions as to how he/she will review the material. We present a second program which reinforces the new information learned in a way which provides the reader with quick, extensive, on-line access to many kinds of help, and allows greater flexibility in learner strategies.

*Language Analysis Project
University of Pennsylvania
440 Williams Hall/CU
Philadelphia, PA 19104

215-898-1955

USING TELECOMMUNICATIONS TO
ENCOURAGE READING AND WRITING

William H. Booz*

Fairfax County Public Schools

The presentation will describe the use of a local electronic bulletin board system (BBS) to foster book reading, creative writing and to provide students with an audience for "publication" of their works. The use of such a BBS in a suburban school system to encourage communition between students, teachers and members of the community on topics of mutual concerns will be discussed.

Telecommunications is a natural extension of productivity software. A nationwide experiment such as The Computer Chronicles Newswire Network enables students to exchange local news stories using telecommunications. Young writers have their own "wire service" to use in creating local newspapers. However, it is possible to have such an idea exchange network locally.

The BBS in Fairfax, Virginia uses bulletin board software, which allows for multiple message and file areas. The Fairfax system has regular discussion areas; however, the BOOKTALK and STRYLINE areas are the ones which encourage book discussion and creative writing. Callers enter messages with their review of a book or react to others' comments. In STRYLINE, callers start stories or continue stories started by others. In addition, teachers can provide their students with an outlet for "publication" of written work by uploading finished stories to be downloaded and read by callers to the BBS. Comments from readers provide the students with useful feedback.

*PDC-Braddock, ES
Fairfax City Schools
7825 Heritage Drive
Annandale, VA 22003

703-750-3842

THE EQUITY IN TECHNOLOGY PROJECT

BRINGING THE BENEFITS OF THE COMPUTER TO ALL STUDENTS

Alice Fredman*

The Educational Computer Consortium of Ohio

The Equity In Technology Project was developed by ECCO in 1985 to address equity concerns in the uses of computers in the schools. This presentation will describe the Project, including planning, implementation and current status.

The Project is a three-year program designed to identify inequities in computer access and devise strategies to alleviate these inequities. It is divided into three phases - Education; Implementation of Local Projects; and Establishment of the Center for Equity in Technology.

Phase 1 - Education, was provided in conjunction with three outstanding national training groups; Playing to Win; Closing the Gap; and Equals in Technology and concluded with the issuing of requests for proposals for local action grants designed to increase opportunities for equitable access to computers.

Phase 2 - Grant recipients will implement their projects. Results of the projects will be the focus of an Equity in Technology Seminar during 1987.

The concluding phase will see the establishment of the Center for Equity in Technology which will serve as a repository for information and resources. The Center will disseminate the findings from the local action grant programs, and serve as a training site for educators and other interested groups.

*1123 S.O.M. Center Road
Cleveland, OH 44124
216-461-0800

SCHOLARLY WEBS: AN ALTERNATIVE EDUCATIONAL SOFTWARE PARADIGM

L. Nancy Garrett, Karen E. Smith

Brown University

Brown University researchers have developed an alternative to conventional educational software that encourages exploration, connectivity and analysis. Intermedia, an integrated multi-media hypertext system, allows authors to link electronic documents on high-resolution graphics workstations. These documents can be created with interactive text, graphics and timeline editors, which are currently part of the system, with plans to develop others. Using Intermedia, students can browse through the resulting webs of information in a non-sequential manner.

By selecting a source document, issuing appropriate menu commands, and then selecting a destination document, authors can create connections between documents to show that a timeline event, for example, is related to more detailed information. In particular, they do this by making a selection of one or more events in a document, issuing the Start Link command, finding the related information in the appropriate document, selecting that information, and finally indicating that there is a connection by using the Complete Link command. Marker icons show the locations of source and destination blocks and the links between them. Students may follow these trails of connectivity by selecting a marker and choosing the Follow command.

Using Intermedia a student in an English literature course could study, for example, a timeline covering Jane Austen's life. The student could follow links to overviews of Jane Austen's novels, historical references, critiques, timelines of authors who influenced her, and even a digitized portrait of this novelist. The student could just explore the trails of this corpus of material or make new connections that might help to study and analyze Jane Austen, or any other author or topic.

*Brown University
IRIS
P.O. Box 1946
Providence, RI 02912

401-863-3639

APPLYING EXPERT SYSTEMS TO CBT

Greg Kearsley*

Park Row Software

Expert system programs represent a new category of applications software that is beginning to have an impact in many different areas. Expert system programs have considerable potential for education and therefore, should be familiar to instructional specialists, especially those involved in computer based instruction.

An expert system is a program that incorporates the rules used by experts to make decisions in a specific domain. The program allows any user to solve problems in that domain by taking advantage of the advice generated by the program. The advice generated is based upon the specific answers given by the user to questions asked by the program.

There are a number of expert system generators available for different computers including inexpensive ones for personal computers. These generators resemble authoring systems and allow someone with no programming knowledge or engineering background to create an expert system. The systems differ in terms of usability, power and rule formats.

Computer based training is a relatively complex form of instruction and one that relatively few managers have much experience with. There is a need to provide considerable guidance to managers to ensure that good decisions are made about CBT projects. The Park Row CBT Analyst is an expert system program that was designed to provide managers with practice making decisions about specific CBT applications. The program provides advice about selecting courses for CBT, selecting a CBT system, estimating CBT development requirements, predicting the success of a CBT project, and evaluating the cost/benefits of CBT.

One of the interesting features of the CBT Analyst is that it includes a rule editor that allows the user to modify, add or delete the built-in rules provided with the program. This permits the program to be "tuned" to local considerations. It also lets students compare the advice given by the expert rules to advice generated via their own rules.

*Park Row Software
1418 Park Row
La Jolla, CA 92037

619-459-2121

METHOD FOR DETERMINING THE APPLICABILITY OF CAI, INTERACTIVE VIDEO, AND TELETRAINING FOR DELIVERY OF INDUSTRIAL TRAINING

Mable B. Kinzie-Berdel*

Arizona State University

Corporate training needs are increasing, both in amount and complexity. More personnel need training in less time and often with fewer training dollars. In order to provide quality instruction in the most effective and efficient manner, many organizations are examining electronic delivery systems to determine their appropriateness for use.

A model has been developed which provides a means of examining the relative applicability of three electronic delivery systems, CAI, Interactive Video and Teletraining, for presentation of existing training courses. The model examines course objectives and current instructional techniques, as well as potential applications of the three delivery systems.

Use of the model results in two things. First, a measure of relative potential of each electronic medium for attaining course objectives is achieved. Secondly, an indication of appropriateness is obtained - do the projected applications for each medium make suitable use of that medium's capabilities? This information is particularly important for an organization's next step: Is projected use of this medium worth what it will cost?

*1711-A E. Hedrick Drive
Tucson, AZ 85719

602-965-4173

THE DESIGN, DEVELOPMENT, AND MANAGEMENT OF A MICROCOMPUTER USERS CENTER

Randal A. Lemke*

Gallaudet College

This case study of two microcomputer user centers analyzes their design, development and management. A microcomputer user center is a centralized place where school personnel and/or students can go to learn how to use computers for their professional/educational needs. One primary need for such a center is that with the recent innovations in use of computers in schools staff members report that they have not been prepared to use the new equipment and have few avenues of learning computer skills and applications.

School personnel can use the center to: 1) Become proficient in the use of professional productivity software for their administrative, materials development, and instructional needs; 2) evaluate instructional software for possible use in classes; and 3) to consult with center staff members for solutions to their unique computer problems. Students also have a need for such a center as it can provide them a place to learn about computers in a non-classroom setting and provide them an opportunity to use the computers for their school work. The centers observed in this study are instructional support units with one serving a large county elementary and secondary school system and the other a small liberal arts college.

***Gallaudet College**
Department of Educational Technology
Washington, D.C. 20002

202-651-5413

EVALUATION OF VIDEODISC-BASED TRAINING TECHNOLOGIES IN A ARMOR MILITARY SETTING

Carl W. Lickteig* - US Army Research Institute

Billy L. Burnside - Training Technology Agency

The US Army Research Institute, The US Army Armor Center and US Army Training and Doctrine Command have established a jointly staffed Training Technology Field Activity (TTFA) at Fort Knox, KY. The mission of this TTFA is to identify, develop, implement and evaluate a variety of low-risk, technology-based training methods and programs. TTFA's initial efforts have focused on training technologies for the M1 tank commander (19K BNCOC). Technologies which result in more effective and efficient training programs for BNCOC will be transferred to other military training programs.

This paper evaluates the TTFA's training programs developed for two separate blocks of instruction: 1) the remedial or prerequisite tasks required for 19K BNCOC enrollment, and 2) a set of land navigation tasks traditionally taught as part of the 19K BNCOC program of instruction (POI). The courseware for both blocks of instruction was delivered by MicroTICCIT System II equipped with a videodisc player interface. A nonequivalent dependent variables (NEDV) experimental design was used to control for Hawthorne effects in general and "computer-experience" effects in particular. This evaluation reports on the effectiveness and efficiency of these training programs relative to the traditional POI and discusses recommendations for future courseware modifications.

*USARI Field Unit
ATTN: PERI-IK, Steele Hall
Ft. Knox, KY 40121-5620

502-624-3450

STRATEGIC PLAN FOR MANAGING TECHNOLOGY IN THE WORKPLACE

Shirley Jean McCann*

Our society has been able to create more communication and information exchanges because of the new technology. The U.S. economy is rapidly expanding from an industrial base to an information service base. An increasing demand for technical competency exists in all career fields. Managers and supervisors are confronted with computerized networks, more technical staff and automated information systems.

However, end-user reluctance to adapt to the increasing rate of technological change affects the impact of technology upon the workplace. The scenarios range from individuals who claim that computers are dehumanizing and inequitable to those who feel that a systematic work environment is impossible without the technology.

Consequently, managerial skills that increase technological acceptance and decrease technological anxieties are essential for more effective use of computerized systems.

The purpose of this session is to present a strategic plan for assessing organizational and individual response to technological change. Management techniques and strategies will be presented for implementing technological change in an organization.

*11640 Deer Forest Road
Reston, VA 22094

703-437-4220

DISTANCE LEARNING FOR IN-SERVICE TEACHERS
VIA MICRO AND MODEM

D. McGrath*, C. DePucchio, C. Franklin, D. Russell
C. Wilson, A. Hayes, S. Kane

University of Illinois

Preliminary observations and findings from the development of a distance learning course for teachers will be presented. The course is the first to be developed in a sequence of courses for teachers in the field who want to take courses in computer education. The first course we chose to implement was Beginning BASIC for Teachers.

The presentation will focus on problems and solutions in CAI and written materials development; keeping the cost to the teacher/school low; equipment problems and decisions; alternatives in the transmission of homework assignments, exams and messages; and pilot testing of the course (not yet implemented--pilot testing will occur over the summer).

The second course to be developed will be sketched out broadly, since the development will begin in September. That course is Computer-Assisted Instruction. It will focus on types of CAI and CMI materials available for instructional use, software evaluation, new trends in CBI, integrating the CAI materials into the curriculum and so forth. The major obstacle in the development of this second course is the availability of demonstration software to the teachers without having to purchase it.

*University of Illinois
Dept. of Secondary Education
1310 S. Sixth Street
Champaign, IL 61801

217-333-0663

AN INTEGRATED APPROACH FOR IMPLEMENTING AND EVALUATING CBI IN A MILITARY ENVIRONMENT

T. M. Shlechter*, D. M. Kristiansen

US Army Research Institute

B. L. Burnside - US Army Training

CBI has frequently encountered difficulties in military training situations. An integrated approach for implementing and evaluating CBI in a military environment has been established by the US Army Armor School, the US Army Training and Doctrine Command and the US Army Research Institute in a jointly staffed Training Technology Field Activity (TTFA) at Fort Knox, KY. The mission of the TTFA is to identify, develop, implement and evaluate a variety of new low-risk technology-based training methods and programs. Many of these new programs are being developed by using computer-based instruction (CBI) in a direct effort to address the need for higher quality training with reduced training resources.

This paper describes TTFA's effort to develop an automated classroom and examines the roles of trainers, training developers, and evaluators in implementing and evaluating this automated classroom. This paper also discusses the unforeseen problems and lessons discovered in this developmental effort. Recommendations are thus made about courseware design standards and implementation and evaluation strategies, especially the need for an integrated approach among trainers, training developers and evaluators.

*USARI Field Unit
ATTN: PERI-IK, Steele Hall
Ft. Knox, KY 40121-5620
502-624-3450

COMPUTER-BASED INSTRUCTIONAL SYSTEMS
UTILIZING ARTIFICIAL INTELLIGENCE

Carroll L. Wilson*

University of Illinois

Several artificial intelligence techniques have been incorporated into new computer-based education systems. What these techniques are and how they work should be understood by educators as well as computer scientists. A good way to gain understanding of these techniques is from the perspective of their use in actual lesson systems. Several such systems are described, which are at various stages of development. Articles about these systems were published in 1985 and 1986

With a clear understanding of actual artificial intelligence techniques used, educators can begin to analyze the tutoring systems for their educational effectiveness. Interesting and exciting ways how some of the new systems achieve their learning objectives are discussed. Even though none of these systems are available for microcomputers, an analysis of their artificial intelligence and educational techniques will provide insights into the new field of intelligent computer-based instruction.

*University of Illinois
Education Building, Rm. 398
Champaign, IL 61820

217-363-6722

EVALUATIONS OF EDUCATIONAL MICROCOMPUTER SOFTWARE:

A CONTENT ANALYSIS OF REVIEWS

Evelyn Bender*

Temple University

A research study is being conducted to determine the criteria used by evaluation organizations for assessing software, the criteria used by periodicals in assessing selected logic/problem solving software, the consistency in ratings of given criteria among published reviews of individual software programs and the relationship of the reviews to the reviewing policies of periodicals.

Evaluation forms used by major evaluation organizations were studied in order to compile a checklist and definitions of criteria. Reviews were obtained of the selected widely reviewed logic/problem solving software programs and a content analysis performed of each. Reviewing policies of periodicals were obtained and compared to reviews. The data is in the process of being summarized and tabluated. Preliminary findings indicate that few of the criteria used in evaluation forms are used in published reviews, and that most ratings of criteria are favorable. Some elements in reviews that render them helpful to selectors are: amount of evaluative vs. descriptive information, evidence of student tryout by reviewer, and mastery of the program by the reviewer.

*7704 Mill Road
Elkins Park, PA 19117
215-634-0357

COMPUTER RESOURCES FOR EDUCATORS

Carol A. Dwyer*

The Pennsylvania State University

Educators who are using computers require information regarding the various resources available to assist in their quest for knowledge about how to use computers in education, reasonably-priced software, reliable and easy-to-access software evaluations, funding for additional purchases and sources for idea-sharing. Many organizations provide educators with reduced-price, inexpensive, or free computer resources such as hardware, software, software evaluations, courses, publications, and help with computer use. However, many educators are not aware of the variety of resources available to them.

Local businesses may be eager to enter into a partnership with educational institutions to share equipment and expertise. Departments within a college or university often develop software to share with K-12 schools. State programs offer funding for software and hardware purchase, discounts to educators and free courses. Services are provided by intermediate units and educational broadcasting stations. Assistance with computer problems may be provided by computer companies, local computer suppliers, user's groups, and educational organizations or agencies. Awareness of the myriad of resources followed by judicious use will promote the integration of computers into the educational process as an instructional, management and problem-solving tool.

*458 Westgate Drive
State College, PA 16803

814-863-9471

LARGE-SCALE CBI WITH MICROS:

WHAT'S MISSING?

Rick Hill,* Leah Rampy

American Express Company

Among the many reasons for using CBI in a large business, perhaps the most compelling is cost-effectiveness. This consideration influences not only the decision to use CBI, but also the selection of the hardware/software vehicle.

Many organizations find it cost-justifiable to start with a small installation and expand to full-scale as warranted. Starting small is a pragmatic approach to getting CBI "in the door" which leads many organizations to opt for microcomputer-based CBI: start-up costs are relatively low and growth can occur in modular increments. However, unless the expansion plan includes a provision to automate management of information among multiple workstations, a cost-effective, small-scale operation could become cost-intensive in the large scale.

What is needed is the comprehensive CMI (Computer-Managed Instruction) component featured in most multi-user CBI systems. This session will explore ways to add this component to heretofore standalone workstations. Authoring and delivery software interfaces will be considered in the context of hardware configurations ranging from local area networks to micro/host links. Also, a CMI system which was conceived and developed at American Express will be profiled as an example of a cost-effective, micro-based CBI solution.

*American Express Company
777 American Expressway
Fort Lauderdale, FL 33337

305-473-3267

SIMULATING COMPLEX PROBLEMS USING DYNAMO

Martin Nyhold*

University of Illinois

This presentation will provide an overview of a version of DYNAMO, a simulation language, that works on microcomputers. Procedures for developing models from description to code, demonstrations of working models, and suggestions for use of the system in the classroom setting will be included. Copies of a simplified version of DYNAMO, for Apple IIs and IBM PCs, will also be available.

The DYNAMO programming language was developed to provide a language for analyzing System Dynamic models. A system is a collection of interacting elements, and the System Dynamics approach is to point out connections between elements and discover how changes in these elements can affect the system as a whole. Problems that involve quantities that vary over time and causal forces that can be quantified are ideally suited for solution using the DYNAMO language. The system provides users with a way of organizing knowledge about problems in a useful but simple format, and it provides the number-crunching support needed to simulate models which vary over time, using varying inputs. DYNAMO has been used to model such varying subjects as the cycle of urban growth and decay, ecosystems, flu epidemics, and the effects of supply and demand on the price of commodities.

*University of Illinois
510 W. Columbia
Champaign, IL 61820

FULL CIRCLE TEST ADMINISTRATION ON MICROS:

INTEGRATED TEST BANKING, SCORING AND GRADING

Judith Pine*

Arizona State University

Microcomputer-based systems are now available which permit the integration of test scoring and item analysis, with a test item bank, and an electronic gradebook. The integration of these components which have been previously used only in a segregated fashion on microcomputers, provides a fully automated system for test administration which offers university faculty flexibility in test design and rapid access to test data in addition to the tremendous savings in clerical time.

Some of the advantages of automatic, integrated systems are apparent in the ability to simulate grade cut-off points; to select test items by codes such as objective, difficulty level, or content; or to rapidly generate multiple versions of tests used in large courses.

The selection of a system for a university department or center must include both technical and logistical criteria such as:

1. Desirable features of each of the software components.
2. Available hardware and the amount of access time necessary for the machine.
3. Long term availability and cost of forms.
4. Potential number of users and depth of use by faculty.
5. Potential clerical cost savings.
6. Need for integration of components.
7. Available personnel to run and learn the system.

*Media Production
Arizona State University
Tempe, AZ 85287

602-965-6738

TESTING 1-2-3: DEVELOPING BENCHMARKS FOR AUTHORING SYSTEMS

Naomi O. Story

Marcopa Community Colleges

As the availability of computers in classrooms becomes more common, educators are wanting to author their own courseware based on their own needs. This is especially true for those in higher education where commercially-produced instructional software is limited in quality and quantity. However, not all faculty have the time or experience to program their own courseware.

Authoring systems have been touted to be the solution to educators' desire to create their own courseware (Merrill, 1985; Pogue, 1984). Basically, authoring systems are considered "user-friendly", and easy to learn. Yet, not all systems can be used by a novice. Systems specifically are designed to facilitate lesson construction. Merrill (1985) says, however, that most systems do not incorporate instructional design principles or are not fully versatile to insure sound instruction.

Faculty do not always have the time, money, or expertise to test different authoring systems. In addition, an authoring system is usually evaluated subjectively by an author while developing a specific courseware. Specific features such as branching capabilities, types of questions, number of responses, or ease of use are tested using a lesson. Hillelsohn (1984) demonstrated an objective process which measured how fast one could author with a system using the same lesson. However, systems were not compared in terms of different types of lesson delivery in an area such as English which is more verbal than visual.

This presentation will describe the development of a variety of benchmarks for evaluating authoring systems. The benchmarks were designed to focus on learner and author needs and different types of lesson delivery such as tutorial, simulation and drill-and-practice used in community college English classes. These benchmarks were used to compare several authoring systems that are designed for the IBM-PCs, Apple IIs, and Macintoshes including McGraw-Hill Authoring System, Instructor, Infowriter, Mentor and Lightsource. Handouts will include the benchmark lessons and a matrix of the comparison results.

*708 W. Bentrup
Chandler, AZ 85224

602-267-4421

WHEN TO USE THE TEAM APPROACH

Margret Hazen*

University of North Carolina

The University of North Carolina at Chapel Hill began an instructional software development project in 1984. Professional staff members (one full-time and several part-time) worked with the twenty-five faculty project members solicited for instructional software development. By 1986 instructional software development was supported by a more specialized team of analysts and designers and other professional staff members (five full-time and several part-time). Several aspects of using professional team members have been found to work exceptionally well , and at other times, a team approach seems to be inapplicable within the confines of institution like the University of North Carolina at Chapel Hill.

When funds are limited, a university may assume that a professional team exists to help a faculty member, but not necessarily to do the work for the faculty member. Several variables have been found to be useful in order to determine the potential usefulness of the team approach for any one project when only a few can be supported well. These include: 1) the faculty member's time and ability to absorb design information or formal critiques; 2) the faculty member's time and ability to organize for sophisticated and thorough evaluation; 3) the faculty member's time and ability to accommodate to sustained curriculum development; 4) the faculty member's time and ability to work with a programmer; 5) the faculty member's need to retain complete control; 6) the need or desire to produce a small local module versus a marketable course; which in turn defines the need to produce documented and maintainable code; 7) the willingness to forego quick development in favor of well designed, high quality production; 8) who supplies the hardware and software for instructional software development.

Reliance upon the faculty members' resources keeps direct costs down, but in many instances it also reduces the amount and quality of the materials that can be produced. Using the team for a few well selected projects allows the development of some high quality, larger volume software.

*University of North Carolina, Chapel Hill
232 Davis Library 080A
Chapel Hill, NC 27514

919-962-0101

AUTOMATING COST/BENEFITS ANALYSIS

Greg Kearsley*

Park Row Software

With the exception of authoring systems, there has been little automation of the instructional development or management process. Analysis, design, and evaluation activities are done almost completely by hand. This makes the development process very labor intensive (i.e. expensive) and subject to human performance limitations and skill differences. To reduce the time and costs involved, as well as increase the reliability, we need to develop automated tools for instructional development and management.

Cost/benefits analysis is a time consuming and tedious activity. The major steps are 1) formulating an appropriate model, 2) identifying the data to be collected, 3) collecting the data, 4) analyzing the data, and 5) drawing conclusions. Automation can help in four of these five steps.

By providing a variety of preformulated models, the initial obstacle to conducting a cost/benefits study can be removed. The Park Row cost/benefits disk contains 10 different models including life cycle, resource requirements, productivity, break-even, compensation and others. The second step can be aided by the generation of data collection forms corresponding to each model. If the data is collected online, a program can be generated that will take the input data and put it in the correct format for the model being used. The data collected can also be analyzed in terms of descriptive statistics.

Of the five major steps, the developer only needs to do the first and last manually, i.e., select the appropriate model and draw conclusions based upon the data analysis. In other words, the two steps requiring human judgement are left to the developer; the routine clerical steps are aided by the software.

A similar approach can be used across all instructional development tasks. A prototype system called the Instructional Toolkit was developed at Courseware Inc. to explore this idea. It uses commercially available applications software such as word processing, spreadsheets and database programs. Practical and theoretical issues raised by the development of this system will be discussed. These include: 1) what can/cannot be automated?, 2) what are the principal benefits of automation?, 3) how can the effects of automation be measured?, 4) what implications does automation have for the skill levels of developers?

*Park Row Software (619-459-2121)
1418 Park Row
LaJolla, CA 92037

HOW LONG DOES IT TAKE?

A DETAILED LOG OF A SOFTWARE DEVELOPMENT PROJECT

Mary Jac Reed*

University of Delaware

A burning question in the minds of managers of CBI and CBT projects is the question of time. How long does it take to develop software materials for the computer?

I was given an excellent opportunity to try to answer this question. I began managing a brand new software project, with new staff using a new authoring language on unfamiliar hardware. The programmers and designers were hired for this project specifically after they agreed to keep detailed logs of their time using a Time Management Utility program. Logs were to document the time devoted to the following activities: learning the new language and the new environment, meeting with coordinators, consultants and subject matter experts, designing screen displays, coding the module, revising the screen displays and the code after student use, after design and code review, and in preparation for the software to be distributed to other institutions.

The project report describes the type of software development project and the findings about the development time. Listings of the times devoted to the various portions of the development process will be available as handouts. The excellent documentation that resulted from the Time Management Utility is useful in forecasting timelines for future projects and in examing ways to streamline the development process.

*University of Delaware
Office of Computer-Based Instruction
305 Willard Hall
Newark, DE 19716

302-451-8161

FACULTY COMPUTING EDUCATION NEEDS:

RESULTS OF THE OHIO STATE UNIVERSITY STUDY

Sallie J. Sherman,*Ajay Bhardwaj

The Ohio State University

A proportionate random sample survey, stratified by college, was conducted as one part of a study to describe The Ohio State University faculty perceptions of their computing education needs and the preferred methods of meeting those needs. Recommendations from the study are being used to guide the implementation of the University faculty computing education program.

A sample of 332 regular faculty was drawn at .05 degree of accuracy. The data sample included 268 (81%) responses.

The survey instrument asked faculty: 1) to evaluate the importance of personally using a computer to perform 53 job tasks and their present ability to do so, 2) to indicate preferences regarding instructional methods and environments, 3) and to provide minimal demographic information.

Descriptive and inferential statistics were used to analyze the survey results at .05 alpha level. The areas of greatest need included research skills (e.g., searches of national computerized databases and the University's computerized library system; creation, revision and conversion of bibliographic entries, and word processing skills (e.g., document creation and revision; typographical, spelling and grammatical checking; manuscript indexing). Immediacy, flexibility and ongoing support emerged as themes for program design.

*333 N. Parkview Avenue
Columbus, OH 43209

614-292-4843

CONTEMPORARY TECHNIQUES FOR CONTEMPORARY STUDENTS: COMPUTER APPLICATIONS FOR PROSPECTIVE MUSIC TEACHERS

Judith Bowman*

Duquesne University

Prompted by the conviction that some facility with music applications of computer assisted instruction is vital in the preparation of prospective music educators, a vehicle for providing this was developed at the School of Music at Duquesne University. The goals were to provide practical acquaintance with music applications of microcomputers and to explore strategies for integrating computer assisted instruction into the music curriculum at several levels and in various subject areas. Implementation was achieved through a series of seminars that constituted part of a grant supported research project. The following procedures were used: 1) Information about the students' prior knowledge of microcomputers and attitudes toward computer assisted instruction was obtained by means of a survey, 2) four seminars were conducted, providing courseware reviews and hands on experience, 3) students evaluated courseware, completed individual projects, and evaluated the seminars, 4) evaluation of students was achieved through observation and individual projects directed toward specific teaching/learning situations.

Seminars were structured by topic. The first session focused on literacy for music educators, music courseware designed for young children, and resources as directories and catalogs. The second session was directed toward courseware appropriate for use in the general music classroom; instructional strategies used in music courseware were studied. A review of the many music theory programs was conducted during the third session; applications both for class and for private studio use were discussed. During the last session students were introduced to management and graphics programs. Specific procedures and applications of computer assisted instruction in the music classroom were discussed.

The project will be continued, flexibility with regard to structure and content is assumed. Practical value and usefulness to participants will be assessed.

*Duquesne University
School of Music
Pittsburgh, PA 15282

412-434-5857

A DISTRICT'S EFFORT FOR CREATIVE USE OF THE COMPUTER IN MUSIC CLASSROOMS

C. A. Brown,* J. Crawford, J. Rehberg

Christina School District

The Christina School District is committed to the utilization of the computer as a tool to improve instruction. This presentation will involve three segments: an overview of the implementation and management of an instructional program involving over 700 microcomputers for 15,000 students; innovative applications in a K-3 music program; and creative aspects of music composition using a microcomputer.

The implementation and management overview will provide methods for providing training and services to teachers in a cost efficient and effective manner. Topics discussed will include: how to provide district support for innovative teachers, central purchasing and distribution of materials and in-district maintenance programs.

The K-3 music presentation will discuss computer-assisted instruction for elementary music concepts and skills and using the computer and a synthesizer for music rehearsals for small instrumental and large choral groups. A video tape will be shown demonstrating the students using the computer in the various application modes.

The 9-12 presentation will emphasize the use of the computer to promote creative applications in the field of music. Three topics will be discussed: creative aspects in music composition; musical arranging; and the physics of sound.

*83 E. Main Street
Newark, DE 19711

302-454-2024

MULTI-FUNCTION COURSEWARE

FOR MUSIC INSTRUCTION

Lan-Ku Chen,* Jon-Chao Hong

National Taiwan Normal University

The courseware contains four primary functions: 1) instructional management; 2) tutorial; 3) drill and practice; and 4) educational game.

The instructional management has the following abilities: 1) save, delete, or collect students' records; 2) teacher or student entry control; 3) date of students' last entry; and 4) student learning path, control.

The tutorial bases on the structure of Robert Gagne's "Events of Instruction". The small segments of the "drill and practice" were inserted in the tutorial when students need additional exercises. The educational game was developed to reinforce the instructional concepts of the lessons.

The courseware includes two music lessons - note recognition and staff and pitch name recognition.

*National Taiwan Normal University
Graduate Institute of Music
Taipei, Taiwan
(02) 9031111

DEVELOPMENT OF A COMPUTER-ASSISTED INTERACTIVE VIDEODISC (CAIV) PROGRAM IN OPERA

Louise Cox*

University of Illinois at Urbana-Chambaign

The purpose of this presentation is to discuss the development and current status of a computer-assisted interactive videodisc (CAIV) program in music education. The program, entitled "What is opera?" uses the opera disc from the Videodisc Music Series, a project recently completed by the University of Delaware that involved the production of a series of eight interactive video-discs for music instruction.

This interactive videodisc program is designed to teach the diverse elements that make up opera; i.e., the roles of the composer, librettist, singing actors, set design, staging, direction, etc. Individualized instruction with immediate feedback is developed to provide a transparent medium to assist students in achieving the objectives of developing an understanding of opera. This CAIV program is intended for use in a music appreciation course.

The talk will describe the following issues involved in courseware design and development: task analysis, content organization, user interaction, screen design and graphic support as well as other development considerations.

*Rt. 150 NW, R.R.3, Box 17
Champaign, IL 61821

217-333-7149

THE MACINTOSH AND CAIM:

DEVELOPMENTS, POTENTIAL, PROBLEMS

Warren E. Hultberg*

Crane School of Music - SUNY

We are currently involved in revising Macintosh systems materials previously developed for use on IBM mainframe and interdata configurations. Languages previously used were Coursewriter, PL/1, FORTRAN, BASIC. Reviewing and/or revising for Macintosh systems materials previously used on Apple II Plus and IIe systems. Languages used on these systems are BASIC and PASCAL; for the Macintosh we are essentially working with the current version of Microsoft Basic. We are preparing new materials designed to maximize the potential of the Macintosh for interactive heuristic program design, combining features of MacWrite and MacPaint. In addition to this basic software for music notation, graphics and sound consideration is being given to utilizing some aspect of certain readily available music programs, most of which are designed for compositional purposes.

The principal paradigmatic design of our new CAI software is based on the concepts of 1) interaction between student and program involving the decision-making process, as solutions to problems are offered by the student or the program; 2) experiencing through hearing, seeing and writing the effects of musical decisions and events as they take place; 3) applying to a variety of practical situations materials being studied elsewhere, e.g., classroom or other CAI programs.

Our current efforts utilize the 512K Macintosh "as is" including as much already available and appropriate software as possible. Future plans involve adding necessary peripherals or other interfaced sound systems.

A typical lab might include standard 512K systems; at least one of which should be configured with an external sound source. At present, although we are concerned with record-keeping and providing hard copy of student responses to the problems presented, these are not among our primary interests. In our view, the CAI lab should provide an environment for musical experiences and study, the effects of which will be seen in enhanced classroom producivity and other activities.

*Crane School of Music
State University of New York
Potsdam, NY 13676

315-267-2413

THE AMIGA AND MUSIC APPLICATIONS

Philip L. McClure*

Cedarville College

In addition to the well-known number-crunching capability that has become synonymous with the computer, the Amiga hardware extensively supports graphic and sound functions which are particularly apt for musical uses. Such features as high-resolution color graphics and animation, a four-channel stereo digital synthesizer, quality text-to-speech synthesis, and sound sampling are built into the machine.

Because of these and other capabilities, a great deal of music software has begun to appear. These programs can be divided into three broad categories (with some overlap): CAIM and authoring systems, score writing and editing, and performance and composition.

The presentation will be a brief overview of the Amiga computer and evaluation of musical packages currently available. Particular attention will be given to applicability for educational purposes.

*P.O. Box 601
Cedarville, OH 45314

513-766-2211, Ext. 549

FEASIBILITY OF INSTRUCTION WITH INTERACTIVE VIDEODISCS CREATED FROM EXISTING VIDEO MEDIA IN INSTRUMENTAL MUSIC EDUCATION

Anne Miller*

University of Illinois at Urbana-Champaign

The cost of producing an interactive videodisc is formidable, so cost efficiency must be achieved through high distribution. By using pre-existing video media, production costs can be reduced, but special problems in making the pre-existing media interactive and educationally sound can result.

The purpose of this study was to (1) develop a videodisc from existing materials designed to teach double reeds. (2) Write software to make the use of the videodisc interactive in order to teach selected cognitive and psychomotor skills to instrumental music education students and double reed students, and 3) evaluate the feasibility of development and use of such an interactive videodisc for music education.

This presentation will address the following issues involved in designing and producing interactive videodiscs from pre-existing video media for use in music education:

1) Needs Assessment - Implications for interactive videodisc technology in music education.

2) Task Analysis - Choosing objectives for music education classes which can be taught using interactive video.
Determining combined uses of motion and sound in the teaching of music which can be taught with an interactive videodisc.

3) System and Courseware Design and Development - Locating available materials which can be transferred to a videodisc format and made interactive in order to teach those objectives identified above.

4) Evaluation - Assessment of student achievement. Cost efficiency analysis.

*2 Patricia Court
Champaign, IL 61821

217-333-7149

EXPLORING POWERFUL IDEAS WITH MUSIC LOGO

J. Kent Williams*

University of North Carolina - Greensboro

Since the advent of microcomputers in the early 1980's, LOGO, the educational programming language, has been widely used as a medium for exploring "powerful ideas" in various subjects, especially mathematics and sciences. MUSIC LOGO, a version of LOGO developed especially for musical activities by Jeanne Bamberger of MIT, has considerable potential as a medium for enhancing the development of significant music-theoretical concepts. This presentation will show how MUSIC LOGO might be utilized and extended in order to explore four "powerful ideas" in music theory: hierarchical structure, melodic diminution, harmonic expansion and variation. By creating the musical equivalent of LOGO "microworlds" for their students to explore, teachers of music theory might more successfully demonstrate the connection between such ideas and actual music.

*University of North Carolina - Greensboro
School of Music
Greensboro, NC 27412-5001

919-379-5911

CREATING INSTRUCTIONAL INTERACTIONS
WITH PILOT PROCEDURES

Gilberto Contento,* Giuliana Cossi

Centro di Ricerca Applicata e Documentazione

A structured program design makes instructional software more effective and efficient. Instructional interactions can be implemented as standard procedures according to simple and/or traditional schemas such as multiple choice, fill-in-the-blank and true/false. Each procedure consists of three main parts: 1) a logical schema which defines the flow of the interaction; 2) a graphical format which determines how the information is displayed; 3) a variant part which substantializes the subject content. The variant part consists of a wide range of parameters including expected answers, feedbacks, number of tries, display time and use of special character sets.

Several modules have been written in SUPERPILOT for the Apple II computer. A distinctive feature is the common format of the data files containing the variant part used by the procedures. Data are compacted to reduce the amount of disk space to a minimum. General subroutines extract data as they are needed. A menu-driven authoring system enables the user, even without knowledge of programming, to generate the variant part of exercises, modify it, manage the data files, pack a number of exercises into a customized test, print tables and analyse student data. The use of the authoring system does not require any predefined instructional strategy. The student can take his own choice among the available exercises listed in ad hoc directories and the teacher can assign a test on the basis of the student name. PILOT programmers can include modules and data files in their lessons and content experts may actively participate in the development of instructional courseware.

*via Pradamano 2/A
Udine, ITALY 33100
432-205643

THE USE OF GRAPHICS IN PILOT PROGRAMS

Lynda Doll*

Georgia State University

The use of graphic images is one of the most popular ways to draw student attention to key information in computer-based instructional materials. Furthermore, research has shown that students have more favorable attitudes toward instruction utilizing graphics.

This presentation will focus on techniques for producing and incorporating graphic images into instruction materials coded in either PC/PILOT or SUPERPILOT. A series of slides will be used to demonstrate a variety of graphic effects. A discussion will follow during which the techniques or tools used to accomplish the effects will be described. Included in this discussion will be the use of the PILOT graphics and special character editors, turtle graphics, the mouse, and several graphics software packages. Ways in which these can be combined to produce animation sequences will also be explored, as will differences in graphics produced for PC/PILOT and SUPERPILOT programs. A handout will provide brief examples of the coding necessary to accomplish several effects.

*530 Sweet Birch Lane
Roswell, GA 30076

404-993-4748

INCORPORATING PC/STORYBOARD INTO PILOT PROGRAMS

Wing M. Fok,* Kern Kwong, Howard Cheung

Georgia State University

The increasing need for CAI has lead to a higher popularity of PC/PILOT. This authoring language requires relatively little programming skills and is easy to use even for people with very little knowledge about computer programming. This language is being used extensively in the education of students in universities and by professional trainers and educators.

However, PILOT is not the best language for all purposes; for example, PILOT provides limited graphic capability. Creating pictures using PILOT requires a comparatively longer amount of time and a higher degree of skill. This limitation can be compensated by the use of PC/Storyboard.

PC/Storyboard provides two libraries of pictures that have already been created. These libraries can be further expanded with the picture-taking feature. Using this feature, the users can capture pictures from any other PC software and display them using PC/Storyboard.

This presentation is on the techniques required to incorporate PC/Storyboard into PILOT programs. A demonstration is also provided to show the result of such combination.

*Georgia State University
Department of Management
Atlanta, GA 30303

404-658-3404

TRANSPORTING PILOT:

A SUPERPILOT TO PC PILOT TEMPLATE

Sharon Gomez*

Harford Community College

There are several ways to transport SUPERPILOT text files to an IBM PC computer. One is to convert SUPERPILOT, based on PASCAL, to DOS; then to use a software package to send the DOS files through an Apple Super Serial Card. The IBM PC, using a third software package, accepts the DOS files. Files can also be transferred with the PASCAL Filer. The Advanced Feature Library of PC PILOT contains a program which "will allow you to copy a file from another system to the PC". Another way is to up and down-load with a mainframe.

PILOT users know that much time is spent in editing. Once SUPERPILOT files are on the PC computer, further editing becomes necessary to enable PC PILOT to obey SUPERPILOT commands. Editing can be significantly reduced by carefully selecting SUPERPILOT commands during initial programming. First, select those commands in SUPERPILOT which are common to PC PILOT. Second, place incompatible commands on separate lines so that a "find and replace" SUPERPILOT request can convert these lines to PC PILOT format. And finally, if you are planning to prepare more than one program,create a template to save time with initial programming and editing after transfer.

*419 S. Union Avenue
Havre De Grace, MD 21078

301-836-4388

SURVIVAL SPANISH INTERACTIVE:

A FIRST-YEAR INTERACTIVE VIDEO COURSE

Fred Jensen,* Mary Ann Lyman

University of Idaho

Survival Spanish, an introductory Spanish video course developed at Miami-Dade Community College, was made "interactive" using the video tape recorder in tandem with the computer. The course was written using SUPERPILOT and has been used successfully at the University of Idaho in one experimental section for a year. A BCD video interface was used to link the two machines together, to code the videotape and to log scenes by video frame codes.

Thirty lessons were designed by Dr. Fred Jensen, who teaches the experimental beginning Spanish section. The lesson structure allows for differing "learning styles" in that the learner may approach the lesson by choosing which portion of the lesson to begin with, or whether or not she/he wants to begin by viewing the whole video in a linear fashion. The student may break the lesson down into smaller sequential steps from the main menu. The student then may opt to approach the smaller portion by viewing the tape or by completing written computer drills. Earlier lessons employed complicated algorithms to analyze the answer string for the presence or absence of certain key features; the later lessons begin to utilize the video capabilities of the authoring language by directing the learner to a video version of the right answer on the second unsuccessful attempt.

Dr. Jensen and Ms. Lyman first discussed the goals of the courseware and defined the population who would find it beneficial. They then drafted a flowchart of the lessons to pinpiont CAI learning-teaching strategy. This helped answer such questions as to what the student should do when he/she has missed a number of successive items, for example. They then designed and tested a modifiable "stamp" to be used for each lesson. The "stamping" allowed for a minimization of errors and reduced considerably the "debugging" time. Each step of the courseware design process is discussed and the software is demonstrated in this presentation.

*University of Idaho
Department of Foreign Languages and Literatures
Moscow, Idaho 83843

208-885-7776

ACQUIRING CONTENT UNIFORMITY ACROSS MULTIPLE SECTIONS OF AN UNDERGRADUATE TECHNICAL WRITING COURSE:

A PROBLEM SOLVED WITH SUPERPILOT

Marilynne W. Mathias,* Elizabeth Ellis

The Pennsylvania State University

The Technical Writing Course is comprised of upper classmen generally enrolled in science and technology majors. The writing of a research paper or a project is a course requirement. A library unit that facilitates the research process involves instruction of information gathering search patterns and use of science and technology information resources. In the past, instruction was delivered to about 800 students by seven librarians who lectured to 40 sections during the third or fourth week of the semester. A task force evaluation of the procedure indicated several areas of concern - among them was a critical need to ensure uniformity of content objectives, increased student involvement, an evaluation component, and a reduction of staff. Several alternatives for improvement were considered. A computer assisted module, using SUPERPILOT, was determined as meeting the requirements.

The module was pilot tested, revised, field tested, revised and is currently in use. Initial findings indicated about 70% of the students achieved the criterion score of 80% or better. Other interesting findings are concerned with time of day and grade point average. Overall, the computer based instruction received high positive student ratings for an efficient method of instructional delivery.

The presentation will include the research findings, a demonstration of the module and the management component, and a question period. Copies of the planning and development documents will be available for review.

*The Pennsylvania State University
303 Wagner Building
University Park, PA 16802

814-863-2271

MICROCOMPUTER DIAGNOSIS OF ARITHMETIC SKILLS

Barbara Signer*

University of Massachusetts/Boston

This presentation will include a demonstration of Computer Adaptive Testing (CAT). The diagnostic programs were authored in IBM PILOT and test the following general mathematics areas: a) numeration, b) computation, c) fractions, d) decimals and percent, e) measurement, f) geometry and g) estimation.

The programs were written to test the high school general mathematics objectives as described in the Boston Public Schools Curriculum Guide. The programs are currently being used in a school with a high rate of absenteeism. Many of the students are older than the traditional high school student and have repeated general math several times. Three classes, currently using the tests as the central structure of the class, are being observed. The purposes of the observations are to conduct formative evaluation to maximize its effectiveness and to observe and interview students to collect affective data. The results of these observations will be discussed.

The tests accept generative responses, that is no choices are given. The use of the random generating function to construct examples allows for multiple test items to test any given objective. Upon completion of the test, results are printed on the video screen, a printer, or saved on a disk. The results show which objectives the student should be studying. Worksheets can then be generated based on the missed objectives.

*45 Carlton Street
Brookline, MA 02146

617-929-8527

EXPERIENCES WITH A SMALL PLATO SYSTEM

Iain Cooke*

Camosun College

A discussion of how a Community College in British Columbia, Canada - where funds for education have been in even shorter supply than most other places on the continent - acquired and operated a small CYBER 810 computer and a PLATO SYSTEM. Since this presentation should be of interest to other sites with (or contemplating the acquisition of) small PLATO systems, the configuration and courseware will be described before proceeding to discuss the College objectives in acquiring the system for a two year evaluation period. Most of the presentation will center on some of the problems encountered in the first year of operation, including those associated with initial training of operations personnel, contractual arrangements with the vendor, as well as some problems encountered in operating the system, including memory limitations, disk capacity, documentation and system management.

The problems associated with persuading college instructors to use the existing courseware and the authoring capabilities will also be discussed. Sprinkled throughout the "problems" are some solutions that have been attempted and even some successes.

*Camosun College
Interurban Campus
4461 Interurban Road, R.R. #3
Victoria, B.C. V8X 3X1 Canada

604-479-7141

STUDENTS AND MAINFRAME AUTHORING SYSTEMS:

STIMULUS FOR DEVELOPMENT OF CBI

Michael Neuman*, Victor Showalter

Capital University

Many small post-secondary institution face complex, apparently insurmountable problems in the development of CBI. Faculty may perceive the advantages of CBI but remain dissatisfied with commercial courseware; they may envision the programs they want, yet hesitate to venture into the development without adequate programming support. Administrators, recognizing the immediacy (and therefore, the primacy) of administrative computing, cannot allocate sufficient support to faculty projects. Students, meanwhile, having been introduced to computers in high school, expect that in college they will use the computer as a tool and not merely as a machine for delivering instruction.

To solve these inter-related problems, the Academic Computing and Development Center of Capital University (using a PLATO system on a CYBER 170/180-815 mainframe), developed a course in which liberal arts students learn the PCD2 authoring system and then collaborate with professors in their major fields to begin the development of courseware that professors can use in their courses. Promising projects are continued by means of stipends to faculty and independent study credit or employment funds for students.

The presentation will survey the problem, the solution, and the costs. The course and its outcomes will be described and the backgrounds and attitudes of students and faculty will be detailed.

*Capital University
Academic Computing
Columbus, OH 43209

614-236-7101

DEVELOPMENT OF PLATO USAGE AMONG FACULTY

Deborah Ouelette*

California State University

The number of faculty and student users on the PLATO system at California State University - Sacramento has increased significantly since it was first installed in 1983. Since that time, on-campus usage has grown beyond 1,200 students and 148 instructors, actively using PLATO for remediation, augmentation of courses, college credit and courseware development.

At the outset, accommodations were arranged for five PLATO terminals connected to the University's CYBER 170-730 mainframe computer in the university computer lab. The nature of the project was exploratory with an overall objective to improve teaching and learning at the university.

With the increasing positive response of both the faculty and students, the PLATO project was accepted and funded. At present, the PLATO learning lab consists of 10 terminals with other smaller satellite labs located in various departments on the campus. In addition software packages have been purchased from other PLATO sites for enhancement of the system along with the implementation of peripheral devices such as sound synthesizers and interactive videodisc.

Recruiting faculty who would provide an organized student usage of PLATO was the key to the successful growth. The means to achieving such growth was through department specific advertisements and workshop/seminars that displayed to the instructors that PLATO would be a time saver and another useful teaching tool that could be utilized with the classroom experience.

Techniques for recruiting, training and stimulating continuous interest for faculty without monetary incentives will be discussed along with special training tools used to minimize the time of the training consultant.

*6516 Deaderick Ct.
Orangeville, CA 95662

916-278-7632

CBE: THE SUCCESSFUL IMPLEMENTATION OF PLATO IN PUBLIC SCHOOLS

Pamela A. Wright*

California State University

California State University (CSU) provides central site PLATO to California Public Schools and has been experiencing highly successful results. The CSU network has the capacity to provide CBE to the Western United States. The current application in public schools is aimed at the truant/dropout population and as a result the program generates its own revenue. The courseware provides a complete continuum of curriculum which integrates appropriately with the California Framework of Curriculum. The courseware is both mastery based and objective based which contributes to the rapid achievement experienced by students. The management system provides for complete track and flow of the student and creates truly individualized instruction.

Recognizing the financial constraints in school districts, computer services have been designed to meet a plethora of educational challenges along with the strategy to generate revenue to fund a PLATO learning center. This concept makes it possible to generate otherwise lost revenue and thus cover the cost of education computer services. The aim is to focus on the truant/dropout population and bring them back into the educational system. Both research and experience have clearly established that PLATO is a highly successful means of retrieving this population which is characterized as being the visual/active learner; the learner who interacts best with technology, but will not perform well in the classroom which is designed for the auditory learner.

In addition to this basis of revenue, other programs can be considered to support computer services. Therefore, it is conceivable that mainframe PLATO can serve a multiplicity of populations with varying needs and this workshop is designed to share the successful models.

*27864 Emerald
Mission Viejo, CA 92691

714-364-1351

THE CENTER FOR SPECIAL EDUCATION TECHNOLOGY

Susan E. Elting* - Council for Exceptional Children

Sharon L. Raimondi - JWK International

The Center for Special Education Technology is a national resource. It collects and exchanges information about using technology with handicapped children and youth. Based at the Council for Exceptional Children (CEC) in Reston, Virginia, the Center is funded by the U.S. Department of Education. CEC is assisted by two subcontractors, JWK International at Annandale, Virginia and LINC Resources of Columbus, Ohio.

The purpose of the Center is to provide information services to teachers, administrators, parents, researchers and others concerned with special education technology. Some priority topics are the selection and use of microcomputer software and assitine devices, equity and consumer issues related to technology, effective strategies for technology-based special education instruction, long-range planning and coordination between home and school.

The Center organizes information in formats that have a high probability of reaching target audiences. Electronic networks, a taped message service and written formats are used to make information easily accessible to educators and parents. The TECH.LINE bulletin board on SpecialNet is designed for teachers and administrators. The board is used as an electronic newsletter and information on a new topic is posted each month.

The Center recently introduced a telephone-based information service called TECH-TAPES. This automated taped message service is available to parents and educators. It is the first nation-wide audiotex service devoted to educational information. TECH-TAPES is available to callers nationwide on the Center's tollfree number.

One very important aspect of the Center's work focuses on research in the use of technology in special education. An annual invitational research symposium brings together active researchers to discuss issues and findings in the field of special education technology.

*Center for Special Education Technology
The Council for Exceptional Children
1920 Association Drive
Reston, VA 22091

703-620-3660

ADAPTATION OF TOOL SOFTWARE TO ASSIST STUDENTS WITH SPECIAL NEEDS

Sheldon D. Rosenberg*

Long Island University

Students who are physically, emotionally and/or learning handicapped require a structured, well-organized and secure learning environment with intensive repetition and opportunity for practice, coupled with immediate feedback. Standardized off-the-shelf, tool software can be adapted to meet these needs. They organize the learning activity into a highly structured format providing the child with a safe and practical instructional system. In addition, these programs offer excellent opportunities for drill, providing a constant instructional base while presenting repetitious educational activities with great patience and consistency.

Preparation and generation of text is basic to the fulfillment of many personal, educational and vocational tasks in our information intensive society. Many students with disabilities that impede their physical and/or intellectual control are unable to use commonly accepted text-generating methods ranging from the simple pen to word processors. In order to mediate certain physical requirements and to support and remediate intellectual weaknesses, we have successfully Beta tested several commercially available utility and word processing programs which have been honed to support a customized educational program.

It must be remembered that computer aided instruction is the most easily adapted method of presenting individualized classroom instruction. Computer System Technology and teacher modified software allow for each student's educational plan to be increasingly responsive and specifically adapted to the individual student's needs.

*110 Livingston Street, Room 808
Brooklyn, NY 11201

718-596-4736

VISUAL IMPAIRMENT AND COMPUTER DISPLAYS:
THE EFFECTS OF FOREGROUND AND BACKGROUND COLOR ON ORAL READING SPEED

Michael John Simmers*

University of Illinois at Urbana-Champaign

Since the passage of Public Law 94-142 and Section 504 of the Rehabilitation Act of 1973 there has been an increased focus on the mainstreaming of disabled students. Mainstreaming implies equal accessibility of educational materials, including computers, by disabled and non-disabled students alike. In spite of the variety of devices becoming available, today's computers vary in their useability by disabled persons.

The needs of the visually impaired (not blind) are neglected more than any other disability group. Thus far, the only consideration given these individuals appears to be the enlargement of the displayed characters. There are many other screen attributes which may interfere with the effective use of computers by these people. One such attribute is the color of the display. This attribute includes both the color of the characters as well as the background against which they are displayed. Until now there has been no investigation involving color displays and the visually impaired. Currently a study is underway which attempts to determine the association between the reading performance of visually impaired students and display color. The presentation will discuss the study, including

1. Who are the visually impaired?
2. What the past research indicates.
3. The procedures for the study.
4. The results obtained.
5. The conclusions and recommendations drawn from the results.

*University of Illinois
Department of Secondary Education
395 Education Building
1310 6th Street
Champaign, IL 61820

*217-333-0227

ARCHITECTURE FOR AN INTELLIGENT TUTOR
USING INTERACTIVE VIDEO AND SITUATIONAL DIAGNOSTICS

A. E. Andrews*, G. P. Lloyd, D. R. Powell, M. L. Stoddard

Los Alamos National Laboratory

Effective intelligent tutors should diagnose student performance deficiencies, provide tutorial remediation, and/or provide learning prescriptions. Interactive video permits presentation of complex situations demanding student response at high cognitive levels (e.g., synthesis, problem solving). These responses, when compared to an expert knowledge base, permit real-time formulation of student specific knowledge models leading to personal tutorial remediation and learning prescription.

The Los Alamos Training Research Team uses a software architecture providing the above features in two PC-based applications. An aircraft recognition tutor helps students learn to identify aircraft in a dynamic environment while a computer-tutor for tactical operations skill development, sponsored by the Army Research Institute, helps students practice battle field skills. The architecture has an expert knowledge base; a controller/driver; and simulation, tutorial, and student modules. Based on student response to the simulation, the student module selects appropriate tutorials and learning prescriptions.

This unique approach focuses on an iterative design methodology and the human-machine interface. The simulation module (using interactive videodisc to present realistic stimulae), with a data collection function, permits formulation of student knowledge representations, supporting the design of the complete tutor's student module. Student feedback is highly individualized being generated from the student and expert modules.

*Los Alamos National Laboratory
P.O. Box 1663
MS F601,S-6
Los Alamos, NM 87545

505-667-6000

INTERACTIVE VIDEO COURSEWARE
AS A RESEARCH TOOL

Janet White Azbell* - Illinois State University

Mary Lou Mosley, Naomi Story

Maricopa Community Colleges

In addition to its use in training and educational applications, computer aided interactive video (C.A.I.V.) courseware also has the potential to become a useful tool for educational research. C.A.I.V. courseware used as a research tool can remedy most of the difficulties associated with the study of instruction through observation of live or videotaped classroom situations. And, with C.A.I.V., the instructional technology itself also becomes a sophisticated research tool for improving the design process, which in turn may be used to improve the quality of instruction that C.A.I.V. can deliver. Research in progress at Maricopa involving the Sony View System and instructional courseware on electronics will be described.

This presentation describes how C.A.I.V. courseware may be designed so that it can 1) provide data on selected research questions, in this case primarily on teacher perceptions and decision-making, and 2) supply information useful in research on the courseware design process itself. The emphasis will be on the design of research, though results of on-going research projects at Illinois State University, Methodist Medical Center of Illinois and Maricopa Community Colleges will be reported as available.

*20601 Fawn Court
Chillicothe, IL 61523

309-672-6795

THE MERGING OF VIDEODISC AND CD ROM TECHNOLOGIES

Greg Baur*

Western Illinois University

In the past several years, a revolution in the technology of data storage and retrieval has taken place. The revolution began with the development of laser optical technology which initially resulted in the videodisc. Though the acceptance of the videodisc by consumers has been somewhat slow, many useful applications have been and continue to be developed.

More recently, another laser optical device has been developed. The compact disk (CD) also knows as CD ROM has received wide consumer acceptance because of its audio reproduction capabilities.

What is CD ROM? How is CD ROM similar to videodisc? How are they different? What is the future of both devices? How will their respective technologies merge?

The presentation will examine answers to these questions. It will also discuss a proposed videodisc/CD ROM project at Western Illinois University.

*Western Illinois University
Department of Computer Science
Macomb, IL 61455

309-298-1939

INTERACTIVE VIDEO IN TODAY'S MARKETPLACE

INDUSTRIAL TRAINING, MANAGEMENT TRAINING, AND POINT OF INFORMATION

Brenda Bracket, Margaret A. Chopp*

Digital Equipment Corporation

This session will enable participants to identify approaches to solving training and information delivery needs using interactive video. Computer-based interactive video (CBIV) has grown from an interesting research and development technology, to an exciting and valuable tool for business and industry. This medium offers the best of all instructional media and more. Yet, the design of effective training and information delivery requires a sense of balance between what can be done with the medium and what should be done. There are many production houses faced with the same challenge. It is critical that CBIV producers present their experiences with this evolving technology, so that as a profession we will become more sophisticated and knowledgeable about maximizing the creative and effective use of this medium.

For the past three years, Adaptive Learning Solutions has offered consulting, and training and information solutions to a wide variety of customers. Using DEC's Interactive Video Information System (VIS), teams of specialists in instructional design, graphic design, video production and software engineering have produced a variety of custom courseware applications. Our presentation will provide the audience with a "mini-tour" of some of those applications. We intend to demonstrate an industrial training course showing the use of software simulations, a management training program outlining a salary and benefits plan, and a point of purchase and information application designed to market a new product in an innovative manner.

---

*30 North Avenue
Burlington, MA 01803
617-273-7492

DESIGN FOR A COMPUTER-INTERACTIVE VIDEODISC MAP ORIENTATION SIMULATION

A. B. Cox*, L. W. Carstensen, J. F. Moore

Virginia Polytechnic Institute and State University

The Department of Geography at Virginia Tech is currently developing a level IV computer-interactive videodisc (CAIV) simulation to aid teaching orienteering skills in an introductory map use course. Orienteering, how we find our way outdoors using maps, is recognized as an excellent method of teaching map use skills. In an orienteering exercise, the participants are asked to locate a series of markers in as short amount of time as possible. The CAIV simulation teaches principles of map use and orientation skills through the use of videodisc surrogate travel.

The CAIV simulation concentrates on two orientation objectives -- position finding and route determination. Learners are expected to 1) accurately locate their positions; 2) locate orienteering markers; 3) make effective and efficient navigation decisions; and 4) accomplish these tasks within a specified amount of time. The computer program will provide options for the instructor to administer the simulation in a variety of ways in order to stress different map orientation techniques.

The benefits of a CAIV orienteering exercise include 1) an opportunity for students to practice before going into the field; 2) a precise and standardized method of instruction for teaching map orientation skills; 3) the elimination of physical endurance and mobility factors resulting in an emphasis on mental abilities and map orientation skills; 4) a standardized method of administering an orientation exercise that will allow for the comparison among students' performances (this has not been possible in the past due to the lack of any method to accurately record the routes taken); 5) the provision of an alternate exercise in the event of poor weather conditions or the absence of an appropriate orienteering area (as in large cities); and 6) the creation of a variety of future orientation environments by changing simulation variables, the computer program, the orienteering map, or by the production of additional videodiscs with different landscapes.

*Virginia Tech
Dept. of Geography
301 Patton Hall
Blacksburg, VA 24061

703-961-7546

CREATING ADAPTIVE LEARNING ENVIRONMENTS
USING EXISTING CBIV COURSEWARE

Russett Ann Craig*

University of Maryland

Responsive, self-contained, adaptive to the learner's needs, with all the potential for high fidelity simulation, computer based interactive video (CBIV) has come to represent the elite in individualized instruction. So much so, that many designers tend to apply CBIV only to such individualized instructional needs.

This presentation will discuss the design, effectiveness, and implications of an experimental "adaptive learning environment" in which groups of adult participants interact with a CBIV program that is only part of a larger learning environment. Three critical variations on the "standard" CBIV theme have been developed to accommodate the learning environment:

1) The focus of attention has been shifted from a specific technology, to a learning environment that utilizes a variety of technologies.

2. An existing and highly engaging stand-alone interactive video program (Decision Points by DEC), has been modified for the purpose of integrating it within a comprehensive workshop curriculum that includes computer analysis tools, a variety of print materials, etc.

3. The role of the instructor has changed from primary information source, to that of a facilitator in a Simulation Environment.

Field test results of this instructional project will be presented along with slides of the instruction in progress, an analysis of the relative strengths and weaknesses of this learning environment, and implications for future inquiry, and design needs.

*University of Maryland
Center for Instructional Development
and Evaluation
College Park, MD 20742
301-985-7365

A PROJECT TO ENHANCE TRAINING VIDEOTAPES WITH COMPUTER-SUPPORTED INTERACTIVITY: LESSONS LEARNED

Doug Farrow* - Logicon, Inc.

Devah Galloway - Courseware, Inc.

This presentation describes one large corporation's initial efforts to convert existing training videotapes to an interactive, computer-supported format. The project team developed interfaces to link existing computers and tape players, courseware to provide the non-video instructional components and a basic student management system to allow performance monitoring and tracking. These new lessons are now being evaluated to determine whether future conversion efforts are cost-justified.

The basic structure of each lesson is as follows: objective, core information, demonstration, practice, scenario and summary. The students interact with the lesson in typical computer-based training format, until they encounter a section of core information or demonstration that is on tape. At that point the display automatically switches from text to video presentation, plays the appropriate instructional segment, and switches back to text once again.

This presentation describes the design and development of the courseware interface and management controls. It focuses upon the lessons learned throughout the project and recommends alternative procedures for future development efforts. The presentation includes a system demonstration.

*Logicon, Inc.
475 School Street NW
Washington, D.C. 20024

202-646-5466

THE DEVELOPMENT OF AN INTERACTIVE VIDEODISC PROGRAM TO DELIVER PESTICIDE APPLICATOR TRAINING IN VIRGINIA

Michelle M. Geasler*, Michael J. Weaver

Virginia Polytechnic Institute

Every three years approximately 15,000 Virginia pesticide applicators are required by the Federal Government to take part in a pesticide applicator training program. Traditional training methods have limited the effectiveness of this training program. In addition, the primary audience for pesticide applicator training is growing as governmental restrictions on the use of pesticides become more stringent. Therefore, the time and money involved in developing and delivering training programs is increasing while personnel and funding resources are not. In order to overcome many of these limitations, the Chemical, Drug and Pesticide Unit, a unit of the Virginia Cooperative Extension Service, at Virginia Tech is developing an interactive videodisc program to deliver pesticide applicator training.

This presentation will describe the decisions and processes involved in the design, production and programming of the videodisc program. The design is based on the "Electronic Book" format developed at MIT, which allows a great deal of textual material to be combined with visuals from the videodisc. Emphases will be placed on the coordination of this textual material with visuals, problems involved in producing a videodisc program of this type, and the methods used to overcome these problems. Results of the pilot program evaluation will be discussed.

*Virginia Polytechnic Institute
Chemical, Drug and Pesticide Unit
139 Smyth Hall
Blacksburg, VA 24061

703-961-6543

AN INTERACTIVE VIDEODISC PROGRAM ON TEENAGE SUICIDE AND DEPRESSION

Carol Hargan*

Lunaria Incorporated

An interactive videodisc curriculum on teenage suicide and depression has been developed under contract from the National Institute of Mental Health to Mandex Incorporated and its instructional design team from Lunaria Incorporated. Research and development support, including production, was provided by the Lister Hill Center of the National Library of Medicine. The effort represents a collaboration between NIMH, NLM, the contractor and a national panel including experts from psychiatry, psychology, nursing, social work, pediatrics and medical education.

The curriculum was initially designed for third and fourth year medical students who plan to specialize in pediatrics and family or general practice. The program, "Suicide Intervention: Assessing Teenagers at Risk", concentrates on developing decision-making skills and knowledge on these major topics: 1) Depression and its relationship to suicidal behavior; 2) interviewing adolescents; 3) assessing suicidal risk; 3) family matters; 4) intervention and follow-up.

The student/users of the curriculum learn about these topics by conducting interviews with four simulated cases, analyzing their data and making decision about diagnosis and treatment. Factual information is interwoven with the cases. All the cases have practical realities such as time constraints, interruptions, attitudes, conflicting demands and limited options for intervention built into the presentations.

The curriculum was pilot-tested in four medical schools in the spring 1986. It is now being revised and will be made available to all interested medical schools in summer 1986. This presentation will discuss the results of the pilot test, describe how the material was disseminated and discuss any barriers to dissemination that were encountered.

*Lunaria Incorporated
2354 King Pl. NW
Washington, D.C. 20007

703-549-3611

COMPUTER BASED INTERACTIVE VIDEOTAPE SIMULATIONS IN PRE-SERVICE TEACHER EDUCATION PROGRAMS

Anne M. Hayes*

University of Illinois at Urbana-Champaign

Pre-service teacher education programs cannot and do not prepare a teacher for his/her first teaching job. Pre-service teachers need experience in identifying problem situations, recognizing available options in those situations, selecting effective strategies and appropriately implementing chosen strategies in interactions with students, school personnel and members of the community. Although some experience is better than no experience, the pre-service teacher also needs to realize that such practice, regardless of its extent, will at best give the learner an incomplete understanding and ability to effectively manage such social phenomena.

Carefully designed computer based interactive videotape (CBIV) simulations can offer pre-service teachers the opportunity to convert methods classroom knowledge into experience if they are properly implemented in the methods classroom.

Pilot results of implementing different versions of a CBIV simulation concerned with managing potentially difficult parent-teacher conferences will be presented in conjunction with demonstrations of the various versions of the CBIV simulation.

*1103 East Scovill
Urbana, IL 61801
217-333-2838

INTERACTIVE VIDEO INSTRUCTION

AND LEARNER CONTROL

Michael C. Hetzel*

Advanced Systems, Incorporated

Successful implementations of Interactive Videodisc Instruction all seem to have one thing in common - they transfer control of the learning situation to the learner.

This presentation looks at learner control as one of the key ingredients to achieving learning compression of up to seventy percent, increased retention over time and greatly increased student motivation.

This presentation has four parts. Part one looks at the theoretical and historical foundations of learner control. Part two discusses unsuccessful implementations of learner control in interactive instruction. Part three demonstrates and discusses a highly successful interactive videodisc instruction product. Part four presents a checklist that can be used to judge the effectiveness of learner control.

*Advanced Systems, Inc.
155 E. Algonquin Road
Arlington Heights, IL 60005

312-981-1500

MATCHING VIDEO TAPE AND INTERACTIVE VIDEO DISC
INSTRUCTIONAL METHODS TO STUDENT'S
FIELD INDEPENDENT/DEPENDENT PREFERENCE

Nancy P. Houston*

Grove City College

Effectiveness and efficiency of interactive video as a method of instruction is an area where more research is needed. Background data on the subject will precede presentation of results of a research study conducted during the spring 1986, semester at Grove City College. This study compares the use of video disc and video tape and attempts to match the student's learning style (field dependent or field independent) to instructional delivery method. Another factor considered is the time taken to complete the material and for outside study.

Students involved in the study will be those enrolled in Computer Introduction and Applications. Nine hours of video disc instruction will be delivered to 45 students; concurrently, 45 other members of the class will view the same material on video tape. Student grade point average and major will be considered in assigning students to the groups to guard against inaccurate results because of intelligence or technical leaning. The Group Embedded Figures Test will be administered to both groups to ascertain field dependence or field independence.

The same tests on the material will be administered to both groups. Scores will be compared and analyzed to determine whether a signficant difference exists. A comparison on time on task and study time will also be made. An attitudinal survey will be administered and the results tallied to determine level of student satisfaction with both methods of delivery.

*615 Superior Street
Grove City, PA 16127
412-458-6600, Etc. 208

STUDENT TOUR OF A VAX 11/785 VIA INTERACTIVE VIDEO

Frederick J. Jenny*

Grove City College

"The students are in the CPU-room and are dismantling the VAX!" That announcement would strike terror into the heart of any system manager of an academic computing center. We would like our students to view PHYSICALLY the components of a computer system that we so often refer to in such computer science classes as Introduction to Computer Organization, Operating Systems and Computer Architecture. But we discourage students from dismantling campus mainframes on their own initiative.

How can they view the "guts" of a VAX? At Grove City College students may "tour" a VAX 11/785 system complete with various disk drives via an interactive videodisc system. Using a DEC PRO 350/IVIS system students can view the VAX from various perspectives and perform such operations such as opening cabinet doors, remove CPU boards, memory boards, load tapes into the tape drive, remove a disk pack, controllers at will, open a RA81 or CDC 9766 diskdrive and view the read/write heads, or view the VAX's synchronous backplane interconnect (SBI). Additional information about each hardware component is available to the student on command.

The design of the system and its innovation application to interactive video will be discussed in this presentation. Simple video animation techniques that simulate user control over the system will be demonstrated.

*Grove City College
Ketler Technological Learning Center
Grove City, PA 16127

412-458-6600, Ext. 208

COMPACT DISC AND VIDEODISC TECHNOLOGY

AN OVERVIEW OF SIMILARITIES AND DIFFERENCES

Carolyn S. Lay*

Vantage Point Systems

The spectacular success of the CD - Audio system (Compact Disc - Digital Audio) in recent years focuses increased attention on related optical disc media, such as the new CD - ROM (Compact Disc - Read Only Memory), CD-I (Compact Disc-Interactive), and the older interactive videodisc. To best understand their respective applications, it is helpful to examine similarities and critical differences between these media.

CD - Audio, first introduced in 1982, has been cited as the most successful consumer electronic product introduction in history. However, the older videodisc, launched with great optimism in 1978, encountered numerous problems during its early days of development. Fortunately, while the learning curve has been costly, the potential of the videodisc is now being realized. The newest forms of optical storage media, the CD - ROM and CD - I, represent the newest steps in the logical evolution of data, audio and video storage. It is likely the development path for these newer technologies will be smoother.

A basic similarity between compact disc and videodisc is their almost identical manufacturing process. Both use microscopic pits to encode information and lasers to read it back. Neither technology presently offers an end user the capability of directly recording information. In contrast, a fundamental format difference exists between these media: compact disc records information digitally, whereas, videodisc records it using an analog signal. Digital data can be easily digested by computers; however, the videodisc/computer combination is complicated by the differences between analog video signals and digital computer output.

In summary, videodisc technology represents the revolutionary form of optical storage; whereas compact disc is the evolutionary form. Because both technologies seem to be converging and absorbing the capabilities of the other, the future for both media is bright.

*Vantage Point Systems
9043 Dunloggin Court
Ellicott City, MD 21043
301-465-6193

A COMPARATIVE REVIEW OF AUDIO METHODOLOGIES IN THE CBIV ENVIRONMENT: NEW DESIGN CAPABILITIES

John F. Moore*, Mary G. Miller

Virginia Polytechnic Institute and State University

Many users and potential users of Computer-Based Interactive Videodisc (CBIV) systems are concerned with audio capabilities and limitations. The market has seen a number of products and methods emerge that provide new options in this area.

Graphic overlay capabilities have been possible for several years. Now, program designers have the freedom to use "voice overlays" to deliver directions, provide remediation, update existing materials, localize generic information, and enrich existing still-frame presentations. Furthermore, clients may now consider voice as an affordable input alternative in certain applications.

This presentation will explore the reasons that expanded audio capabilities will be an important component in the future CBIV environment. We will compare design flexibility, quality, capacity and cost for the following products and methods: Generating voice from ROM; IBM voice assistant board; VOTAN voice board; encoding audio as videodisc frames; external magnetic storage; CD-ROM. Using a videotape case study, we will demonstrate the design flexibility obtained when programmable voice synthesis and voice recognition input are included in the CBIV environment. User reactions from system field trials will be discussed.

*Virginia Tech
Learning Resources Center
Blacksburg, VA 24061

703-961-5879

GETTING INVOLVED IN INTERACTIVE VIDEO:

PROBLEMS AND PITFALLS

Terry A. Olivier*

Emporia State University

Recent technological innovations have led to the development of a new and modern instructional technology called interactive video. Interactive video effectively combines instructional television (ITV) and computer assisted instruction (CAI). The combination of these two instructional media has the potential for creating a very powerful teaching and learning tool.

The major advantage of interactive video is that it creates a learning environment that capitalizes on the power of instructional television and computer-assisted instruction, while limiting the disadvantages of both.

The educational community is rapidly becoming aware of the potential of interactive video as an educational tool. A recent survey by the Great Plains National ITV Library indicated that nearly 20 percent of K through 12 schools and colleges now own videodisc playback equipment. In the report submitted to the Congress by the Office of Technology Assessment it was estimated that by the end of 1985 more than 250,000 interactive videodisc systems would be in educational use in the United States.

How can universities provide instruction in interactive video technology in a time of growing need when campus budgets are once again under the ax? Are interactive video production costs effective for a small university? This presentation will cover advantages and disadvantages as well as comparative costs of interactive video equipment. The presentation will also discuss the problems and pitfalls a small university faces when it decides to embrace interactive video technology.

*Emporia State University
1200 Commercial Street, Box 36
Emporia, KS 66801

316-342-0900

BEYOND TAPE VERSUS DISC: ISSUES IN THE DESIGN OF INTERACTIVE VIDEO

Timothy L. Phillips*, Michael J. Hannafin

The Pennsylvania State University

One of the current areas of concern in the development of instructional interactive video involves the issue of which video delivery medium is most appropriate for use with this relatively new technology. Most of the concerns involve the two most commonly used types, namely videodisc and videotape, and the advantages and limitations of each. These concerns are very technocentrically based, that is allowing the capabilities of the technology to be the central issue in determining the design of an instructional medium. Whereas, the central concern should be focused on the needs of the recipient of the instruction, namely the learner and his capabilities and limitation. Therefore, the purpose of this paper is to review empirical research involving interactive video and other related research based upon learner capabilities.

Properties of interactive video assumed to be of significance in terms of improving learning shall be examined based upon empirically-based findings and principals. The instruction for purposes of this paper shall be broken down into five phases and the impact of interactive video's capabilities on each shall be examined. These phases include orientation, presentation, sequence, encoding and retrieval. In addition, an empirically-derived perspective for the design of interactive video, one which derives design features from within a comprehensive cognitive framework is presented.

*The Pennsylvania State University
310 Rackley Building
University Park, PA 16802
814-865-6560

OVERLAY GRAPHICS FOR INTERACTIVE VIDEODISC SYSTEMS

D. V. Pigford*

Western Illinois University

One option to fully utilize the potentiality of interactive videodisc systems is the use of overlay graphics. This technique allows the superimposing of computer generated graphics over the standard NTSC videodisc image. The result is a workstation that uses one monitor for displaying both source images.

The importance of this technique, from a designer's perspective, is that information from the videodisc can be "updated". An old sale price can be superimposed with a new one from the computer graphics system. New information, figures and highlighting can be added to the application in real time even though the disc has been mastered. This pseudo-update technique is invaluable in helping a system to expand or to hide errors in videodisc images.

The technique has the disadvantage of technical complexity. One problem is flickering or scintillation which can be distracting and undesirable. The other problem of resolution, computer memory, and speed can vitally effect the success of the intended application.

Incorporating the overlay option in a microcomputer-driven videodisc system can be accomplished in various ways. One can purchase a complete system with the overlay graphics included. Other options are to purchase special adapters, software, and mixer/keyed devices. The various approaches, along with product information, will be given during the discussion of overlay graphics.

*Western Illinois University
Department of Computer Science
Macomb, IL 61455

309-298-1452

IBM INFOWINDOW SYSTEM: A NEW WORLD OF INTERACTION AND EXCITEMENT FOR VIDEODISC

Marguerite Raaen*

IBM/Georgia State University

Whether INFOWINDOW is used to train or educate, to inform or to sell, its presentation power lies in the impact of the media elements it uses...and in how easily you can combine the elements for exciting and involving multimedia presentation. With these technologies and Video PASSAGE, the new authoring system that IBM announced, you can present full-motion pictures, stills, touch screen or keyboard interaction, and narration or sound from the two audio tracks. Graphics in up to 16 simultaneous colors from a palette of 64 can be displayed or overlayed. The audio chip lets you include synthetic speech and sound effects. PASSAGE allows you: to include complex "branching" techniques, to accept graphics and text from a number of IBM graphics packages, to specify touch points on the INFOWINDOW display, to create synthetic speech to guide a student or user, and to include still and/or motion sequences from a videodisc. INFOWINDOW PILOT, a newly announced authoring language, can also increase your application productivity and allow you, the designer, to incorporate videodisc technology with your PC for a more dynamic presentation. Various generic videodisc software for training and education will be available for this new system. The Adult Reading package from NEC is now available and previewed in this session along with two new authoring systems and IBM's new videodisc system, INFOWINDOW.

*IBM Corporation
219 Perimeter Center Parkway, Suite 450
Atlanta, GA 30346

404-399-1518

A BARGAIN BASEMENT GENERIC
INTERACTIVE VIDEODISC

Mark Sanders*

Virginia Polytechnic Institute and State University

For the past three years, the author of this presentation has been exploring inexpensive means of incorporating interactive video in educational settings. This work has led from interactive videotape to a generic disc concept. This presentation will report the current status of a project aimed at producing a low-cost generic interactive videodisc.

The first stage of the project began with the solicitation of slides and/or slide-tape instructional programs from faculty throughout our College of Education. These slides (totalling about 4000) address a variety of content areas from teaching methods to robotics. Once accumulated, these slides were transferred very inexpensively to 16mm film by our Learning Resources Center. Next the film was transferred to videotape and then to videodisc.

The second phase involved the merger of voice simulation with this videodisc. In an attempt to keep costs down, voice simulation was utilized rather than the more expensive Still-Frame-Audio technology.

The final phase involved designing the computer code to provide interactivity. The nature of the video (i.e. still-frames) lent itself to the retroactive development of this computer code, thereby keeping the developmental costs down.

*Virginia Tech
144 Smyth Hall
Blacksburg, VA 24061
703-961-6480

THE DESIGN AND EVALUATION OF INTERACTIVE VIDEO DISC LESSON IN CHEMISTRY

S. G. Smith, L. L. Jones, M. L. Waugh*

University of Illinois at Urbana-Champaign

This presentation will describe the development and evaluation of a series of interactive videodisc lessons designed to teach concepts in general chemistry. An IBM PC computer, a MicroKey interface card and a Pioneer LD-V1000 video disc player have been combined to create an instructional medium which uses the distinctive capabilities of these separate technologies to deliver highly interactive instruction.

These lessons are largely tutorial and simulation in nature. Most involve simulated laboratory experiences which involve the students in hypothesizing, experimenting, collecting data, calculating and analyzing data. The lessons are highly branched, allow for variation in student input, and provide instructional feedback responses. In addition, the lessons provide instantaneous access to high quality video images, and combine these images with computer generated text and graphics overlays to focus students on the interpretation of experimental results and encourage additional experimentation.

Four separate evaluation plans have been developed to assess the effectiveness of these lessons. Students' attitudes and perceptions of the lessons will be measured by an end of course questionnaire. The influence of the lessons on student achievement will be assessed in three different ways. The lessons will be examined for use as a possible lab replacement, lab enhancement and pre- or post-organizer for lecture content.

*University of Illinois
Department of Secondary Education
386 Education Building
1310 S. Sixth Street
Champaign, Illinois 61820

217-333-3676

THE ROLE OF THE COMPUTER IN SECOND LANGUAGE ACQUISITION RESEARCH

Dana Boatman*, Christopher Cieri, Nadine O'Connor

University of Pennsylvania

This paper will demonstrate the potential utility of a computer in second language acquisition research. Two major considerations in experimental testing are the control of test variables and the accumulation and manipulation of relevant test data. The design and data from two tests will be compared. We will discuss the advantages of using the computer in the administration of a written test.

We hypothesize and provide examples to support the claim that a computer program which 1) controls for the format, amount and type of test input and output; 2) provides the researcher with information not available in standard test-taking environment; 3) avoids individual test-taking difficulties which may arise; 4) determines test-taking strategies, and 5) provides standardized on-line input for analytical tools, will produce research both qualitatively and quantitatively superior to traditional methods.

*University of Pennsylvania
Language Analysis Projects
440 Williams Hall/CU
Philadelphia, PA 19104

215-898-1955

THE USE OF EXPERT SYSTEMS AND SEMANTIC NETWORKS IN THE EXPLANATION OF PHYSICS PRINCIPLES TO STUDENTS IN AN INTELLIGENT CAI SYSTEM

Stephen F. Brown

Texas Learning Technology Group

During the course of learner-based instruction using a knowledge-based tutor, the student periodically reaches key moments during which a meaningful response from the system can provide that unique statement which ties concepts or principles together into the student's cognitive structure. To accomplish this, the system must not only possess the expert system necessary for complex explanations, but must also have a robust student model capable of drawing the correct portions of the expert knowledge into the appropriate verbalizations.

The system described here (ARPHY II) is a modified version of the intelligent physics tutor presented at ADCIS in 1985. The expert knowledge consists of production rules governing the physical phenomena associated with accelerated motion. Forward and backward chaining provide the means to allow the tutor to reason through a problem, being capable of answering basic "why" and "how" questions, while frame-based knowledge associated with each principle is attached to the appropriate nodes. A networked hierarchy of types of learning types guides the possible paths of instruction. A genetic graph structure allows a student model to be developed which keeps track of student acquisition of knowledge, particularly with respect to temporal development, deviation and correction of skills, and abstraction of principles. The focus of the discussion will be on the development of the genetic graph and the operation of the module responsible for communications between the student and expert modules.

*Texas Learning Technology Group
P.O. Box 400
Austin, TX 78767-0400
512-476-9116, Ext. 414

THE CRAFTY ART OF USING ANALOGIES

Kathy A. Fletcher*, Janis Morariu, Marcia Whitney

The University of Maryland University College

An analogy, or similarity between like features of two things on which a comparison can be based, can be an effective strategy for computer-based instruction (CBI). By linking new learning to concepts already understood by the learner, analogies can make abstract ideas more concrete and also lend structure for the active assimilation of new ideas. In CBI, analogies may be visual or textual depending upon the type of information being represented.

Using analogies appropriately is an art, requiring instructional designers to become crafty artists. Some precautions must be considered. Since not all features of analogous examples are shared with the new body of information, it is possible for the learner to over-generalize from the analogy to the new information. If the learner does not have previous experience with an analogous example, then instruction may be unduly complicated since the learner will have to struggle with understanding the example as well as the new information.

Despite these pitfalls, recent research lends support to the premise that analogies are effective under certain conditions and for certain kinds of subjects. CBI is an ideal medium for using and evaluating the effectiveness of analogies in instruction.

*University of Maryland University College
Center for Instruction, Development
and Adult Education
University Blvd. & Adelphi Road
College Park, MD 20742

301-985-7356

RELATIONSHIPS BETWEEN CBI
AND RESEARCH ON TEACHING

John Guthrie, Mark G. Gillingham*

University of Maryland

The components of computer-based instruction (CBI) have matured during the past decade to the point where their instructional benefits can be tested and compared with components of traditional instruction. A meta-analysis of published CBI since 1980 is reported. Instructional components gleaned from research on teaching (attention/motivation, presentation of knowledge, guidance, and assessment) were compared to both the observed instructional components of CBI and to components purported by expert designers of CBI. Results indicate that most CBI studies do not adequately report data, and those studies that do, report large effect sizes. However, studies that report the largest effect sizes do not compare instructional components of an equal nature or size. In addition, except for assessment, there is great overlap between the instructional components gleaned from research on teaching and the components purported by CBI experts. The importance of studying critical instructional components of CBI is emphasized. It is suggested that CBI has three goals, development, media design, and instruction, each of which can make use of quality research on the critical components of instruction.

*University of Maryland
Department of Curriculum & Instruction
College Park, MD 20742

301-454-5590

THE IMPACT OF COMPUTER USAGE ON UNDERSTANDING MATHEMATICAL OPERATIONS

W. Hannum*, H. Ballew, J. M. Gleason, J. Kolb, C. Stewart

University of North Carolina at Chapel Hill

This presentation reports a study that compared learners who used a computer as a problem solving tool in mathematics with those who used calculators to solve the same problems. Seventh grade students were randomly assigned to the computer or calculator group to study a lesson on operations with fractions. Both groups practiced solving the same problems but the computer group used a specially designed software package that allowed the user to specify which of several operations to perform in solving the problem thus freeing them from the need to do many calculations. For example, the student could select a routine that found the greatest common multiple or lowest common denominator for a problem. Thus, they could focus on selecting and sequencing the specific operations necessary to solve the problems rather than having to repeat many laborious calculations by hand. This may allow them to use higher order cognitive processes more associated with problem solving.

Results indicated no statistically significant differences in terms of ability to do calculations with fractions ($t=-0.6295$, $p > .05$), speed of calculations ($t-0.4387$, $p > .05$) but did indicate statistically significant differences in terms of understanding and ordering the operations involved ($t=-2.1297$), attitudes towards mathematics ($t=4.84$, $p < .05$) and towards computer usage $t=3.04$, $p < .05$). Retention data are being collected and will be included in the presentation.

These results are discussed in terms of the cognitive consequences of using problem solving software. It appears that problem solving applications may facilitate a deeper understanding of mathematical content without penalizing basic computation skills. Implications for theory and further research are made.

*University of North Carolina at Chapel Hill
School of Education
Chapel Hill, NC 27514

919-962-3791

THE EFFECTIVENESS OF DIFFERENT RESPONDING APPROACHES IN MULTIPLE-CHOICE OF CAI DESIGN

Jon-Chao Hong*

National Taiwan Normal University

The implementation of the computer in education has become an essential tool in instructional settings. Differing from the conventional way, computers used in multiple-choice design allow students to employ many approaches. Furthermore, it is assumed that different ways to access the learning materials would result in different outcomes of learning.

In addition to the conventional one, this study draws from three approaches: inversing, underlining, and scanning off to act upon those characters appearing on the line of the screen to test which would be the most effective among the types of responding approaches. The result of this study indicates that the "inversing" approach is the most effective, in general.

*National Taiwan Normal University
Department of Industrial Education
Taipei, Taiwan R.O.C.

02-321-9742

THE EFFECTS OF COMPUTER-BASED KNOWLEDGE REPRESENTATION STRATEGIES IN THE ACQUISITION OF SELECTED MICROECONOMICS CONCEPTS

Vincent E. Lasnik*

The Ohio State University

A study currently in progress at The Ohio State University examines the issue of how instructional design theory and cognitive learning theory can be synthesized in the development of a computer-based unit on selected basic concepts in microeconomics. Applying research on concept formation, individual differences, learner control, motivation, and cognitive psychology generally, two macro-level knowledge representation strategies were designed to investigate the efficacy of inductive (i.e., problem-centered, guided discovery learning of an inferential network of interrelated concepts) and deductive (i.e., rule-based, highly structured learning of hierarchically-organized concepts) process approaches to the teaching of basic concepts (i.e., supply, demand, equilibrium price and quantity) in microeconomics.

Component Display Theory and Elaboration Theory were employed to guide the construction of the micro and macro level instructional design components of the two treatments. The differential effects of each strategy will be measured in relation to individual differences and performance on standardized instruments of economic literacy achievement (i.e., Test of Economic Literacy; Jr. High School Test of Economics).

Inductive sequencing includes an EGRULE (example-rule) concept formation strategy, in addition to a motivational problem scenario to enhance deep-processing, and the derivation of principles within a network of coordinate and probabalistic interrelationships (i.e., Bayesian logic). Deductive sequencing is syllogistic and employs a RULEG strategy to confirm explicit given principles. It is hypothesized the inductive strategy will be superior for some students, and the deductive strategy for others. The updated findings of the study will be presented with the appropriate theory base, methodology, treatments, results and suggestions for future research in computer-based instruction.

*2333 Neil Avenue, No. 4
Columbus, OH 43202

614-422-4872

THE APPLICATION OF A MODEL FOR DESIGNING INTERACTIVE AND RESPONSIVE COURSEWARE

Cindy B. Leshin*, Vernon S. Gerlach

Arizona State University

Although there are many models for designing and developing effective instruction there are few existing guidelines or models for the design of computer-based training. Computers have received a great deal of attention as a new technology with the capability to improve the effectiveness of instruction. However, much of the training designed for CBT has failed to incorporate two essential attributes. Whatever else, good computer-based training must be interactive and responsive.

A model has been developed that provides specific prescriptions for designing interactive and responsive instruction. An instructional designer who was trained in the use of the model wrote specifications and developed storyboard screen sheets which contain minimum instructions for encoding a simple lesson on triangles. An authoring language was used by graduate students to develop an instructional program from these screen sheets.

The resulting instructional program will be evaluated on the basis of existing criteria used for the development of effective instruction. Since this model has previously been used to produce highly effective programmed instruction in text form, it is expected that the use of this prescriptive model will result in the effective CAI that is both interactive and responsive.

In order to test this hypothesis, the computer-based instructional program on triangles will be tried out with fifth graders to assess how effective it is. If the results are positive, further experiments will be conducted to test the generalizability of the model to other kinds of instruction and other programmers/designers.

*6101 North 51st Place
Paradize Valley, AZ 85253

602-965-6738

LOGO STYLES AND ABILITY IN
7TH GRADE STUDENTS

Lucy B. Mallan*

C.E.S. Jewish Day School

Research is beginning on differences in programming style attributable to ability. An opportunity occurred to explore LOGO programming styles in about ninety 7th graders. Five homogeneously grouped math classes, learning disabled to honors, produced LOGO programs in groups of 2-4 children. The study correlates measures of LOGO sophistication (number of procedures, length of procedures, use of recursion and of color), with standardized test scores.

It is expected that brighter students will have more sophisticated programs, by these measures, but it is interesting to speculate on whether the relationship is linear. Some differences may be present by sex.

Although the presenter herself taught all five classes, teaching styles were inevitably different because of the wide variations among the groups. Importance of procedural thinking, however, was stressed in all groups. Preliminary conclusions show ability correlated with number and brevity of procedures, but not with use of color.

Though criteria for measuring creativity as opposed to sophistication are by definition nebulous, it is believed that the better math students have more creative programs as well as more sophisticated. To demonstrate this actual programs created by the children will be presented along with statistical evidence.

*3235 38th Street NW
Washington, D.C. 20016

301-881-1404

REPORT ON UCLA LOGIC SOFTWARE PROJECT

Robert L. Martin*

System Development Corporation

Version 1 of LOGIC: A WORKBOOK tests "non-intrusiveness" as a design philosophy for instructional software. Preliminary results indicate agreement with (Munro 1985) on the success of this approach. The software supports the learning of the formal language of first order logic, symbolization of English sentences, practice on applying rules of inference, and a natural deduction proof system (Kalish and Montague). The proof-checker, set in a full-screen editor for constructing and revising derivations, is non-intrusive — feedback is provided when the student requests it. More general respects in which the software is non-intrusive: 1) it supports, rather than replaces, the course lectures by enabling students to do homework at the computer with correction and explanation - printouts of corrected assignments are provided to TA's giving them more time for teaching; 2) the user has full control of movement throughout the program.

*System Development Corporation
2525 Colorado Avenue
Santa Monica, CA 90406-9988

QUALITATIVE RESEARCH:

WHAT CAN IT BRING TO CBE?

Delia Neuman*

University of Maryland

Increasingly, researchers are citing the need to acquire more in-depth information about the ways in which courseware is actually used during instruction. The development of maximally effective CBE materials ultimately depends upon an understanding of the interactions of students, teachers, and courseware within the rich and complex environment of the classroom.

The qualitative, or naturalistic, paradigm offers the most promising methodology for gaining the detailed, context-bound information required. Receiving renewed interest in educational research, this approach entails an intensive investigation of the inter-relationships of a variety of factors. In terms of CBE, the paradigm focuses on individual students' reactions to specific dimensions of the courseware they encounter in day-to-day instruction and addresses specific ways in which indivudal students and teachers interact with courseware in order to learn and to teach.

This presentation discusses the most significant assumptions underlying naturalistic inquiry and details the methods of data collection and analysis associated with the qualitative approach. Drawing upon an extensive study of the use of over 20 commercial courseware packages by seven teachers and over sixty students, the presentation demonstrates the kinds of insights this methodology can offer to courseware designers and the ways in which those insights can be obtained and verified.

*University of Maryland
College of Library and Information Services
College Park, MD 20742
301-454-5441

ICAI: INTELLIGENT APPLICATION OF AI

Ok-choon Park*, Robert J. Seidel

U.S. Army Research Institute

Artificial Intelligence (AI) technology provides a powerful means for the development of highly adaptive CBI systems. However, the practical value of current ICAI systems (in terms of instructional effectiveness and efficiency) seems to be seriously limited because of the insufficient incorporation of instructional theories and principles in the development process.

This presentation has two main purposes: 1) to stimulate combined and cooperative efforts between researchers/developers in AI and Instruction; 2) to provide conceptual guidelines for the development of future ICAI systems. Specific issues to be discussed in this presentation include the following:

1. Historical development of CBI from programmed instruction to ICAI;
2. Components of ICAI and AI methods for developing the components in the system;
3. A review of representative ICAI systems in terms of the AI methods used to develop the instructional components;
4. Important differences between ICAI and CBI systems;
5. Syntactical differences and semantic implications of ICAI and CBI terminologies;
6. A model of adaptive instruction for the development of future ICAI systems;
7. Specific recommendations and research issues for the development of future ICAI systems.

*U.S. Army Research Institute (PERI-II)
5001 Eisenhower Avenue
Alexandria, VA 22333

202-274-8829

ADAPTIVE EXTERNAL PACING AS AN ALTERNATIVE TO SELF-PACING IN CAI

Kyle Peck*

NBI, Inc.

This study, to be completed by July 1986, employs the computer to deliver instruction and to determine and adapt to individual differences in cognitive processing rate. By doing so it becomes possible to evaluate the merits of "internal pacing" (within the student) and "external pacing" (provided by the computer) without contamination caused by forcing students to process information at inappropriate rates. In this study four rates of individualized external pacing are compared to conventional methods of learner control and "group-based" external pacing.

The subjects will be 140 sixth grade students randomly assigned to seven experimental conditions: 1) a "baseline" condition, 2) a self-paced condition, 3) a group-based externally-paced condition, 4) an individualized externally-paced condition, 5) a moderately faster individualized externally-paced condition, 6) a significantly faster individualized externally-paced condition, and 7) an individualized externally-paced condition where cognitive processing time is added.

Students in all seven groups process information on introductory screens in self-paced mode. Time spent on each screen is recorded, averaged and compared to the baseline group to determine individual processing rates. Individual rates and experimental conditions combine to determine the time each display remains visible.

Conditions will be compared on post-test achievement, efficiency (a measure of achievement per minute) and attitudes concerning the lesson.

*NBI, Inc.
P.O. Box 9001
Boulder, CO 80301
303-938-2625

THE EFFECTS OF PRACTICE AND ORIENTING ACTIVITIES ON LEARNING IN AN INTERACTIVE VIDEO LESSON

Timothy L. Phillips*, Michael J. Hannafin, Steven M. Tripp

The Pennsylvania State University

The instructional utilization of interactive video has greatly increased in recent years. However, the dominating influence effecting applications of interactive video appear to be based on the capabilities of the technology. Very little empirical research has been performed based on the learner capabilities to either support or refute the current applications. Therefore, there is a need to perform empirical research to assist in the development of appropriate guidelines for the design of interactive video.

In this study the effects of practice and orienting activities on the learning of cued and uncued factual information and application skills were studied. Sixty undergraduate students were presented with an interactive video lesson on art history utilizing the National Gallery of Art videodisc. Students received either a cognitive orienting activity or a prompt to utilize their own orienting skills prior to each lesson section. Following each instructional segment, students were presented with one of three levels of practice. The practice levels ranged from no practice to an elaborate practice. Based upon prior research several hypotheses have been developed: first, that level of practice will be the dominating factor in learner ability to recall information; second, that there is a direct correlation between level of practice and learner recall; and third, as level of practice decreases the effect of orienting activities will increase.

*The Pennsylvania State University
310 Rackley Building
University Park, PA 16802

814-865-6560

THE EVOLUTION OF COMPUTER-BASED INSTRUCTION:

BEHAVIORAL TO COGNITIVE

Lloyd P. Rieber*, Michael J. Hannafin

The Pennsylvania State University

Computer-based instruction (CBI) has been influenced largely by the behavioral notions of learning stemming from the programmed instruction (PI) era. This stimulus-response (S-R) foundation, based primarily on the work of B. F. Skinner, implies that learning is the result of external forces acting upon the learner. However, recent research seems to point to some clear advantages to reconsidering this perspective to take into account the role of the learner during learning. This approach, regarded as a cognitive orientation to learning, highlights and emphasizes the importance of this learner role.

The presentation will begin by summarizing many of the hallmarks of the behavioral and cognitive schools in order to show the evolutionary process which has occurred apart from CBI research. The primary purpose of this presentation, however, is to share a cognitive framework for the design of computer-based instruction. This framework is specifically derived largely from Robert Gagne's Events of Instruction. This framework will join together many of the cognitive theories milestones into a working model by which CBI designers can base instructional decisions.

*The Pennsylvania State University
302 Rackley Building
University Park, PA 16802
814-863-2427

MODELS FOR THE DEVELOPMENT AND DESIGN OF CAI:

AN ANALYSIS OF COMPREHENSIVENESS AND OPERATIONAL LEVEL

Robin Taylor*

University of Maryland Baltimore County

Previous methods for analyzing models for the development and design of instruction (Andrews & Goodson, 1980; Stamas, 1973) have weaknesses that limit their usefulness for identifying optimal reference models. The weaknesses stem from artificially separating tasks into "development models" and "design models", using very broad task specifications, and ignoring models' purposes.

To overcome these weaknesses, a new analysis process has been devised. Its major features are:

1. From the literature, determining the major tasks involved in the development and design of instruction (consisting of twelve tasks common to the development of any instructional materials, and six peculiar to CAI) and the subtasks they include.
2. Analyzing each model for only those major tasks it addresses.
3. Assigning a measure of importance (priority rating) to each subtask, and using priority ratings to determine how completely a model addresses a task (comprehensiveness rating).
4. For each task, determining which models are usable as the sole reference source for a novice in the field (operational level rating).

Using this method of analysis, a set of four models has been identified which, taken together, are maximally comprehensive for the six CAI tasks (Chambers & Sprecher, 1983; Drotter & Roth, 1984; Moyer, 1984; Steinberg, 1984). This set of models also constitutes the maximally operational reference set.

*UMBC
Department of Education
5401 Wilkens Avenue
Catonsville, MD 21228

301-455-2305

VISUAL METAPHOR AS AN INSTRUCTIONAL STRATEGY

Steven D. Tripp

The Pennsylvania State University

Numerous researchers have investigated the physical dimensions of visualized instruction (color, detail, etc.). Salomon, among others, has investigated the symbolic aspects of media. It is possible to take a middle ground and investigate the symbolic aspects of visual images. In spite of Knowlton's suggestion in 1966 and earlier that "analogical pictures" may have potential for instruction there appears to have been little interest in this line of research. This may have been, in part, due to the relative paucity of psychological literature which would have served as theoretical underpinning for such a line of research. In recent years a minimal body of such research has emerged along with a resulting concensus that metaphor is a cognitive phenomenon, not a linguistic one.

Given such support, it is now possible to test empirically the instructional effects of visual metaphor and offer a rationale for those results. This presentation will discuss the beginnings of a line of research into the use of visual metaphor with microcomputer-based instruction.

*1131 S. Pugh Street
State College, PA 16801
814-863-1192

ADCIS Proceedings, November 1986

PHYSICS EXPERIMENTS ON INTERACTIVE VIDEO DISCS

Ronald Gautreau, Harpreet Chawla,
Jenhaw Chang, Jihn-Fa Jan,
Shuenn-Tyan Lee, William Savin,
New Jersey Institute of Technology
Newark, New Jersey 07102

ABSTRACT

We will show various Physics laboratories that we have developed around level 3 Interactive Video Discs. The unique and innovative laboratories employ computer generated overlayed graphics which allow students to make interactive physical measurements directly from the monitor screen, and then interactively analyze the data taken.

INTRODUCTION

The well known features of Interactive Video Discs (IVDs) make them a natural medium to use in education in general, and science education in particular. In this presentation we will show various level 3 IVD Physics laboratories that we have developed for use in our freshman and sophomore Physics courses. This is a continuation of level 1 Physics experiments that we reported on at the 27th ADCIS meeting (Gautreau, 1986).

The laboratories are built around sequences from two commercially available video discs--"Astronomy" from the Space Disc series published by Video Vision Associates (now Optical Data Corporation), and "The Physics of Automobile Collisions," by Robert Fuller and Dean Zollman, that is published by J. Wiley, Inc. The software for the level 3 interactivity was developed using VAX Producer and is delivered on Digital Equipment Corporation's IVIS system.

We first give a detailed description of one particular laboratory to illustrate the innovative interactive features we have employed for presenting the subject material and making physical measurements. Other laboratories will then be described.

AUTOMOBILE COLLISION EXPERIMENTS

One IVD laboratory we have developed makes use of sequences from "The Physics of Automobile Collisions" to illustrate the physical principles of impulse and momentum during collisions. When a body of mass m undergoes a one dimensional collision with another object during a time interval Δt, the impulse exerted by the force acting on the body causes the momentum of the body to change, which can be described by

$$(1) \qquad F_{avg} \Delta t = mv_f - mv_i$$

where v_i and v_f are the velocities of the body before and after the collision, and F_{avg} is defined by Impulse = $F_{avg} \Delta t$.

The particular video sequence on which the experimental part of the laboratory is based shows a slow motion video picture of a car crashing into and rebounding from a wall. The experiment consists of the student measuring the velocity of the car before the collision, v_i, the velocity of the car after the collision, v_f, and the time interval Δt that the car was in contact with the wall. The student then uses equation (1) to determine the average force F_{avg} exerted on the car during the collision with the wall.

The velocity measurements are accomplished by the student's measuring the position of the car as a function of time, and then determining the velocity from the slope of a position vs time plot. The time is displayed by means of a graphically overlayed clock coupled to the video disc frame numbers.

There are several ways we have developed for measuring the car's position. In one method, the student, using the computer keyboard, advances the video picture of the car to a convenient location, where the image of the car is left frozen. The student then uses the function keys to manipulate a graphical pointer until the student has determined that the pointer is correctly superimposed on a mark on the video image. At the option of the instructor, we have developed two different ways of locating the image. The student can press a defined key, and the location of the car will be automatically recorded from a computer program that correlates the pixels on the video monitor with a scale that has been calibrated previously by the student.

Alternatively, the graphical pointer will be superimposed on a graphical distance scale that the student has calibrated previously, and the student will read the scale and enter his data into the computer. In either case, the student must make a judgement about superimposing the graphical pointer on the video image, which is the essence of actual experimental data taking.

An alternative way we have used for measuring the car's position is to superimpose graphically a grid on top of the video image of the car, with the grid being calibrated by the student. The student then uses the computer keyboard to advance the video image of the car stepwise, and enters the grid reading at each location into the computer. With this procedure, the student must make a judgement as to the location of the car relative to the superimposed grid, just as would be done using measuring rulers in a laboratory.

When the student is finished taking the distance vs time data, a distance vs time plot of the data he has just taken is displayed. The data should correspond to a straight line, and, using the monitor's touch screen, the student draws on the plot what he considers to be the best fit straight line. The computer then calculates the slope of this line, which is equal to the car's velocity.

The time Δt that the car is in contact with the wall is measured by advancing the video image of the car until it just begins to contact the wall, starting the clock, and stopping the clock when the car just looses contact with the wall after the collision. In this case, the student has to make an experimental judgement about when the car just contacts and looses contact with the wall. The experimental result for Δt is entered into the computer.

After the student completes the measurements and enters the experimental values for v_i, v_f, Δt, and the mass m of the car into the computer, the computer then calculates the average force F_{avg} exerted on the car during the collision from equation (1), thereby completing the experiment. It is important to recognize that the procedures for making measurements in our IVD car crash simulation are actually the same type of measurements that a professional scientist would make if he were asked to analyze a real car crash.

The experimentation part described above is one section of a complete stand alone IVD Physics laboratory about impulse and momentum. When the student enters the laboratory, he can go anywhere to the following main menu items:

1. Registration,
2. Object and Preview,
3. Theory,
4. Procedure,
5. Experimentation,
6. Data Analysis,
7. Calculations,
8. Results and Conclusions,
9. Applications.

Thus, if a student is entering the laboratory for the first time, he would probably want to see the Object and Preview, whereas if the student has already taken the data in the Experimentation part, he might want to go to the Data Analysis or Conclusion section. Upon going to a particular section, the student has choices from a sub menu of selections from that section. The combination of main menu and submenu in each of the sections offers flexibility in a student's choice of where he wants to go in the laboratory. As a bit of an entertaining feature, the main menu and each sub menu is accompanied by music from the audio track of the video disc that the student can listen to as he is making his menu choice.

Each section is developed around images from the video disc, with appropriate superimposed computer generated color graphics. For example, the Theory section shows a video sequence of a car crashing into a wall. As the car crashes into the wall, an arrow overlayed at the front of the car shows the buildup and decrease of the force exerted on the car by the wall during the collision. At the same time, a computer generated overlay shows an F vs t plot of the buildup and decrease of the force. Along with this there are text and equations describing the physical concepts of momentum equalling the mass times velocity of the car, and impulse being the area under the F vs t curve. The use of video and audio, along with color graphics makes the presentation much more effective and understandable than standard computer aided instruction that uses computer graphics alone.

There are other IVD Physics laboratories that we have developed along the lines described above. Using another sequence from "The Physics of Automobile Collisions" video disc showing two automobiles colliding at right angles, sticking together, and going off with some common velocity, our students verify the conservation of momentum principle given by $\sum mv_f = \sum mv_i$. In this experiment, the students measure the inital and final velocities of each of the cars before and after the right angle collision.

GRAVITATIONAL ACCELERATION EXPERIMENTS

Video Vision's "Astronomy" video disc has three consecutive sequences around which we have built laboratories measuring gravitational acceleration. The first sequence shows a man dropping a hammer and feather on the earth. This is followed by an astronaut dropping a hammer and feather on the moon. The final sequence shows an astronaut "dropping" objects in an accelerated space ship.

For each video sequence, the student is shown the frames on which he is going to make measurements. The student is given the option to view the sequence frame by frame in any order he chooses. As each frame is shown, the student is given the time the objects have been falling to that point, and a measure scale is overlayed. At the same time two graphs are superimposed on the video, one for graphing distance versus time, and theh second for graphing distance/time versus time.

The student uses the curser movement keys or the touch screen to measure the position of the falling objects. Once the student is satisfied with the measurement, he goes on to another frame, and the data is automatically recorded on the graphs. As measurements are made on each frame, the student sees the data collect on the graphs. If any point looks bad, the student can redo the measurements on that frame.

When the student decides that he has enough data, he then can go on to the Analysis of Data section. In this section, the student is given full screen displays of his data, with the option to print out the data and/or the graphs. Since the distance-time relation for a constantly accelerating body with initial velocity v_0 is given by

$$d/t = v_0 + \frac{1}{2}gt \qquad (2)$$

one expects a distance/time versus time graph to be a straight line with a slope equal to half the acceleration of gravity. The student chooses interactively by eye a straight line to fit the data. The line chosen is shown on the graph and the student is allowed to change the fit until he is satisfied. The slope and intercept are then calculated by the computer. The student is also offered the opportunity to do a linear regression analysis on the data to determine the slope and intercept.

The other gravitation video experiments are done in a similar manner. In all the experiments, the student makes measurements and interprets a unique set of data. The feeling that each experiment is really under the student's control, rather than a computer demonstration, is unique to these IVD lessons. As one becomes proficient with the IVD hardware, the experiments take on a real world feeling, and the students come to feel as if they have done a hands-onexperiment.

In this manner, the students can interactively measure the acceleration of gravity on the earth, on the moon, and in an accelerated reference system. While it is possible to measure the gravitation acceleration of a body on the earth in Physics laboratories by other means, the measurement of the acceleration of gravity on the moon or in an accelerated rocket ship is of course not possible in earth based laboratories.

CONCLUSION

The coupling of computer graphics with audio and video images from video discs provides an exciting medium for illustrating physical principles in the classroom and laboratory. To our knowledge, the methods we have developed for using the IVD medium to generate overlayed graphics for students to make actual physical measurements of situations of interest in Physics are new. The interactive data taking, data analysis, and reaching conclusions about physical quantities allows students to employ directly the scientific method they are taught in the classroom. This constitutes a new innovative use of the "Interactive" concept of Interactive Video Discs.

ACKNOWLEDGEMENT

We gratefully acknowlege a SIG grant from the Digital Equipment Corporation that made this work possible.

REFERENCE

Gautreau, Ronald (1986). Simple Interactive Video Disc Experiments on Motion in Physics. Proceedings of the 27th International ADCIS Conference, p. 168.

Contact: Ronald Gautreau
New Jersey Institute of Technology
Newark, NJ 07102

201-596-3556

PROCEEDINGS INDEX

A

Page

Abder, Pamela F. ... 347
Adler, Alan G. ... 406
Aljadir, Leta P. ... 411
Anbar, Michael ... 1
Andrews, A. E. ... 464
Arnold, Jean ... 401

B

Bailey, M. Gene ... 116
Baker, Patti R. ... 269
Ballew, H. ... 490
Bastow-Shoop, Holly E. ... 412
Bauer, Carol A. ... 352
Baur, Greg ... 466
Beadenkopf, F. Scott ... 403
Belland, John C. ... 269
Bell, Sally A. ... 8
Bender, Howard J. ... 353
Bender, Evelyn ... 432
Berlin, Donna F. ... 347
Bhardwaj, Ajay ... 441
Birkel, Lane F. ... 385
Blombach, Ann K. ... 174
Blumenfeld, Gerald J. ... 28, 386
Boatman, Dana ... 420, 486
Booz, William H. ... 421
Bork, Alfred ... 387
Bowman, Judith ... 442
Bowers, Steve ... 81
Boyer, Linda A. ... 388
Bracket, Brenda ... 467
Braune, R. ... 381
Brown, Stephen F. ... 231, 487
Brown, C. A. ... 443
Burger, Michael L. ... 33
Burnside, Billy L. ... 427, 430
Burston, Geoff ... 354

C

Carr, Victor ... 373
Carr, Norm ... 354
Carstensen, L. W. ... 468
Carter, Tracie ... 201
Chang, Jenhaw ... 503
Chawla, Harpreet ... 503
Chen, Thomas T. ... 146
Chen, Minder ... 111
Chen, Lan-Ku ... 444
Cheung, Howard ... 452
Cheyer, J. ... 413
Chimura, Hiroyasu ... 100
Chopp, Margaret A. ... 467
Cieri, Christopher ... 420, 486
Clark, W. Bruce ... 95
Clegg, Victoria L. ... 355
Colamaio, Maryanne E. ... 249
Conklin, George H. ... 122
Conkright, Tom D. ... 5

Page

Contento, Gilberto ... 450
Cooke, Iain ... 457
Cossi, Giuliana ... 450
Cox, A. B. ... 468
Cox, Louise ... 445
Craig, Russett Ann ... 469
Craig, Elisabeth M. ... 236
Crawford, J. ... 443

D

Dalton, David W. ... 241, 275
Dassance, Richard A. ... 245
Davis, S. C. ... 356
Davis, T. ... 380, 381
Dear, Brian L. ... 357
DePucchio, D. ... 429
Doll, Lynda ... 451
Dwyer, Carol A. ... 433

E

Eddins, John M. ... 279
Ellertson, J. D. ... 358
Ellis, Elizabeth ... 455
Elting, Susan E. ... 461
Emerson, Joe D. ... 332
Entwistle, B. ... 381
Erickson, Jeffrey B. ... 369
Ezell, Margaret P. ... 149

F

Fahs, M. C. ... 410
Farr, Beatrice J. ... 8
Farrow, Doug ... 359, 470
Ferrero, Grace W. ... 389
Field, Terry S. ... 402
Fletcher, Kathy A. ... 488
Fok, Wing M. ... 452
Franklin, C. ... 429
Fredette, Barbara W. ... 389
Fredman, Alice ... 422

G

Galloway, Devah ... 470
Gardner, Patrick R. ... 360
Garrett, Nancy L. ... 423
Gautreau, Ronald ... 503
Geasler, Michelle M. ... 471
Gerlach, Vernon S. ... 493
Gillespie, Lori ... 361
Gillingham, Mark G. ... 489
Gleason, Jane Marie ... 390, 490
Glowinski, Debra J. ... 125
Gomez, Sharon ... 453
Goodwin, Stephanie S. ... 362
Grimm III, Floyd M. ... 391
Gross, Phil ... 236
Gulick, Elsie E. ... 401
Guthrie, John ... 489

PROCEEDINGS INDEX

H **Page**
Hackler, Nadine 414
Handler, Barbara H. 363
Hannafin, Michael J. 59, 249, 308, 481, 499, 500
Hannum, Wallace H. 390, 392, 490
Hargan, Carol 472
Harrell, Cherie 395
Harshbarger, Susan 415
Harvey, Francis A. 283
Hawkins, Ruth M. 364
Hayes, Anne M. 429, 473
Hazen, Margret 155,438
Hazen, Thomas Lee 155
Heaton, Linda 414
Heines, Jesse M. 11, 393
Hettinger, Bonney 365
Hetzel, Michael C. 474
Hewitt, A. N. 366
Higgins, Norman 367
Hill, Rick 434
Hofstetter, Fred T. 179
Hong, Jon-Chao 288, 368, 444, 491
Horine, Julie E. 369
Houston, Nancy P. 476
Hultberg, Warren E. 446

I
Israelite, Larry 11

J
Jan, Jihn-Fa 503
Jeger, Abraham M. 105
Jelden, David L. 370
Jenny, Frederick J. 477
Jensen, Fred 454
Jensh, Ronald P. 403
Jonassen, David H. 392
Jones, L. L. 485
Jorczak, Robert L. 371
Just, Steven 372

K
Kalnitsky, George 72
Kane, S. 429
Kapp, Ann B. 416
Kato, Hiroshi 100
Kearsley, Greg 424, 439
Kern, Richard 332
Kinzie-Berdel, Mable B. 425
Kline, Judith S. 417
Koehler, Tommas J. 128
Kolb, J. 490
Kristiansen, D. M. 430
Kurata, Masahiko 100
Kuznetsov, Helen 208
Kwong, Kern 452

L **Page**
LaBrosse, C. 366
Lakshminarasimhan, A. L. 135
Lambiris, Michael A. K. 214
Larson, Donna E. 76
Lasnik, Vincent E. 492
Lawson, Patricia 15
Lay, Carolyn S. 478
Lee, Shuenn-Tyan 503
Lemke, Randal A. 426
Leshin, Cindy B. 493
Leslie, S. M. 159
Levin, James A. 394
Lewis, Elaine 292
Lickteig, Carl W. 427
Lippert, Renate C. 407
Lloyd, G. P. 464
Locatis, Craig 373
Lommel, Jane M. 355
Losleben, David R. 92, 382
Lowe, J. 410
Lucado, Scott R. 5
Lucking, Robert A. 395
Lyman, Mary Ann 296, 454

M
Mallan, Lucy B. 494
Malloy, Jane A. 36
Mane, A. 381
Martin, Robert L. 495
Mathias, Marilynne W. 396, 455
McCann, Shirley J. 428
McClure, Philip L. 447
McGrath, D. 429
McMullen, David W. 40
Merrill, John A. 374
Miller, Anne 448
Miller, Edith A. 419
Miller, Mary G. 375, 479
Miller, Richard V. 391
Millikin, Elsie G. 301
Mock, Valerie E. 165
Moore, C. N. 376
Moore, John F. 468, 479
Moore, Victoria 404, 405
Morariu, Janis 19, 488
Morimoto, Yasuhiro 100
Morrison, James L. 84
Mosley, Mary Lou 465

N
Neuman, Delia 496
Neuman, Michael 458
Nunamaker, Jay F. 111
Nyhold, Martin 435

O
Olivier, Terry A. 480
Olson, Michael J. 256

PROCEEDINGS INDEX

Page

Ouelette, Deborah 459
O'Connor, Nadine 420, 486
O'Neil, Daniel K. 406
O'Neill, A. 366

P

Park, Ok-choon 497
Peck, Kyle 498
Peters, J. T. 410
Phillips, Timothy L. 481, 499
Pigford, D. V. 482
Pine, Judith 436
Popplewell, Harold E. 256
Powell, D. R. 464

Q

Quinby, B. 413

R

Raaen, Marguerite 483
Raimondi, Sharon L. 461
Rampy, Leah 434
Reed, Mary Jac 142, 440
Rehberg, J. 443
Reinhart, Richard 377
Richard, Jerry 378
Rieber, Lloyd P. 308, 500
Roberson, E. Wayne 125
Roberts, Nancy 46
Robinson, Steven L. 397
Rosenberg, Sheldon D. 462
Ruby, Ralph 314
Russell, D. 429
Rust, Grosvenor C. 379

S

Sabol, Mark A. 8
Sanders, Mark 484
Sato, Takahiro 100
Savin, William 503
Scherer, Charlotte 53, 398
Schnippert, William W. 221
Seidel, Robert J. 497
Sherman, Sallie J. 441
Shlechter, T. M. 430
Shneiderman, B. E. 19
Showalter, Victor 458
Shtern, Victor 292
Signer, Barbara 456
Simmers, Michael J. 463
Sisler, Grovalynn 418
Sivasankaran, Taracad R. 314
Slotnick, Robert S. 105

Page

Smith, S. G. 485
Smith, Karen E. 423
Smith Shaw, Doris 324
Spannaus, Timothy W. 22
Starkweather, John A. 187
Stewart, C. 490
Stoddard, Mary L. 332, 464
Stokes, A. 380, 381
Story, Naomi O. 437, 465
Suchinsky, Adele S. 92, 382
Sullivan, D. 356
Swadener, Marc 59
Sweeters, William 383
Szabo, Michael 225, 354

T

Tatsuoka, Kikumi 338
Taylor, Robin 501
Thomas, Barbara 81
Tripp, Steven D. 499, 502
Trollip, Stanley R. 407
Tuthill, G. Steven 264

V

Vaughn, Gary 399
Vispoel, Walter 343

W

Wang, Y. Richard 111
Waugh, Michael L. 66, 394, 485
Weaver, Michael J. 471
Weller, M. 381
Welsh, Charles V. 146
Wesswick, Louise 419
West, M. 356
West, Elizabeth 400
White Azbell, Janet 465
White, Arthur L. 347
White, Charles S. 408
Whitney, Marcia 488
Wickens, C. 380, 381
Wiggins, Emily S. 417
Will, Paul J. 168
Williams, J. Kent 449
Wilson, Caroll L. 429, 431
Wilson, Lois S. 8
Worman, Scott 409
Wosar, William A. 225
Wright, Barbara J. 384
Wright, Pamela A. 460

Y

Yens, D. P. 410

Printed by: **OMNIPRESS** . . . **"The Proceedings Printer"**
Madison, Wisconsin